

## Air Force Institute of Technology AFIT Scholar

---

Theses and Dissertations

Student Graduate Works

---

12-1-2018

# Real-Time Heuristics and Metaheuristics for Static and Dynammic Weapon Target Assignments

Alexander G. Kline

Follow this and additional works at: <https://scholar.afit.edu/etd>

Part of the [Operational Research Commons](#)

---

### Recommended Citation

Kline, Alexander G., "Real-Time Heuristics and Metaheuristics for Static and Dynammic Weapon Target Assignments" (2018). *Theses and Dissertations*. 1942.

<https://scholar.afit.edu/etd/1942>

This Dissertation is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact [richard.mansfield@afit.edu](mailto:richard.mansfield@afit.edu).



**Real-Time Heuristics and Metaheuristics for  
Static and Dynamic Weapon Target Assignments**

DISSERTATION

Alexander G. Kline, CPT, USA

AFIT-ENS-DS-18-D-016

**DEPARTMENT OF THE AIR FORCE  
AIR UNIVERSITY**

***AIR FORCE INSTITUTE OF TECHNOLOGY***

**Wright-Patterson Air Force Base, Ohio**

DISTRIBUTION STATEMENT A  
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENS-DS-18-D-016

Real-Time Heuristics and Metaheuristics for Static and Dynamic Weapon Target  
Assignments

DISSERTATION

Presented to the Faculty  
Graduate School of Engineering and Management  
Air Force Institute of Technology  
Air University  
Air Education and Training Command  
in Partial Fulfillment of the Requirements for the  
Degree of Doctor of Philosophy in Operations Research

Alexander G. Kline, MS  
CPT, USA

December 2018

DISTRIBUTION STATEMENT A  
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENS-DS-18-D-016

Real-Time Heuristics and Metaheuristics for Static and Dynamic Weapon Target  
Assignments

DISSERTATION

Alexander G. Kline, MS  
CPT, USA

Committee Membership:

Dr. Darryl K. Ahner, PE  
Chair

Dr. Brian J. Lunday  
Member

Dr. Daniel W. Steeneck  
Member

Dr. Meir N. Pachter  
Member

Dr. Adedeji B. Badiru  
Dean, Graduate School of Engineering  
and Management

## Abstract

The problem of targeting and engaging individual missiles (targets) with an arsenal of interceptors (weapons) is known as the weapon target assignment problem. This problem has been well-researched since the seminal work in 1958. There are two distinct categories of the weapon target assignment problem: static and dynamic. The static weapon target assignment problem considers a single instance in which a known number of incoming missiles is to be engaged with a finite number of interceptors. By contrast, the dynamic weapon target assignment problem considers either follow on engagement(s) should the first engagement(s) fail, a subsequent salvo of incoming missiles, or both.

This research seeks to define and solve a realistic dynamic model. First, assignment heuristics and metaheuristics are developed to provide rapid near-optimal solutions to the static weapon target assignment. Next, a technique capable of determining how many of each interceptor type to reserve for a second salvo by means of approximate dynamic programming is developed. Lastly, a model that realistically considers erratic flight paths of incoming missiles and determines assignments and firing sequences of interceptors within a simulation to minimize the number of hits to a protected asset is developed.

Additionally, the first contemporary survey of the weapon target assignment problem since 1985 is presented. Collectively, this work extends the research of missile defense into practical application more so than currently is found within the literature.

*To my children, wife, and parents.*

## Acknowledgements

There are many people to whom I owe a great deal in the completion of this effort. First, I would not have had the opportunity to come to AFIT without the help of LTC Lee Evans and Dr. Brian Lunday. Thank you for the academic and professional guidance you have shared with me over the past three years.

I want to thank my classmates for helping me through the difficulties of this program with constant upbeat optimism and support. Thanks for all of your help, guys.

I want to thank my family for their patience and support even when my attention was stuck at school. The process has not been easy but you have been my source of energy and motivation throughout.

I want to thank my parents for teaching me that no obstacle is insurmountable and for believing in me even when the odds were against me.

Lastly, I want to thank my advisor, Dr. Darryl Ahner, for his guidance and encouragement over the past three years. It was his recommendation which afforded me this amazing opportunity and his achievement which inspired my constant effort. Thank you for continuously illuminating the unknown, sir. I hope to have earned this.

Alexander G. Kline



# Table of Contents

	Page
Abstract .....	iv
Dedication .....	v
Acknowledgements .....	vi
List of Figures .....	x
List of Tables .....	xi
I. Introduction .....	1
II. Literature Review .....	7
2.1 Introduction .....	7
2.2 Formulations .....	9
SWTA Formulations .....	10
DWTa Formulations .....	17
2.3 Exact Algorithms .....	29
SWTA .....	30
DWTa .....	31
2.4 Heuristic Algorithms .....	33
SWTA .....	33
DWTa .....	35
2.5 Discussion .....	37
Evolution of WTA .....	37
Recent Developments .....	38
Alternate Applications .....	41
Analysis of Literature Influence .....	41
2.6 Conclusion .....	43
III. A Heuristic and Metaheuristic Approach to the SWTA .....	45
3.1 Introduction .....	45
3.2 Quiz Problem Heuristic .....	47
3.3 Eminent Domain Metaheuristic .....	51
Improvements on the ED Metaheuristic .....	56
3.4 Computational Results .....	58
Comparison of Heuristics .....	59
Improvements with ED Metaheuristic .....	60
Optimality Gaps of Heuristics .....	65
3.5 Conclusion .....	67

	Page
IV. Transforming the Weapon Target Assignment Problem for Efficient Optimal Convergence .....	72
4.1 Introduction .....	72
4.2 WTA Reformulations .....	74
4.3 Testing and Results .....	80
4.4 Conclusion .....	83
V. A Greedy Hungarian-like Algorithm for the SWTA .....	85
5.1 Introduction .....	85
5.2 QP Heuristic and ED Metaheuristic .....	86
QP Heuristic .....	86
ED Metaheuristic .....	87
5.3 Greedy Hungarian Heuristic .....	88
5.4 Computational Results .....	92
5.5 Conclusion .....	97
VI. Implementing the CAVE Algorithm for the Heterogeneous 2 Stage DWTA .....	98
6.1 Introduction .....	98
6.2 Literature Review .....	99
6.3 Solution techniques .....	100
CAVE Algorithm .....	101
MDP .....	105
Baseline Policy .....	108
6.4 Computational Results .....	108
Small Experimental Design .....	110
Large Experimental Design .....	113
6.5 Conclusion .....	113
VII. A Continuous Time Two Stage Shoot-Look-Shoot Weapon Target Assignment Problem .....	116
7.1 Introduction .....	116
7.2 The Model .....	118
7.3 The Solution Techniques .....	123
A Single Stage .....	123
Two Stages .....	124
Baseline Policy .....	126
7.4 Computational Results .....	126
One Stage Problem .....	129
Two Stage Problem .....	131
7.5 Conclusion .....	133

	Page
VIII. Conclusion .....	135
Bibliography .....	141

## List of Figures

Figure		Page
1	Quiz Problem Heuristic and Modified Quiz Problem Heuristic .....	50
2	Eminent Domain Metaheuristic .....	53
3	Slope-based ED and Ratio-based ED Assignment Value Decay Rates for 2 Instances Each of 4 Different Problem Sizes .....	62
4	Relative Improvements Using Multiple Simultaneous Denials in the Slope-based ED Metaheuristic (with $p_a = 0.4$ ) .....	63
5	Greedy Hungarian Heuristic Flow Chart .....	89
6	Greedy Hungarian-like Heuristic Flow Chart .....	102
7	3-D Model using Bézier Curves .....	122
8	One Stage Procedure .....	125
9	Probabilities of Hit/Kill Distributions .....	128

## List of Tables

Table	Page
1	WTA Literature by Citation Density ..... 42
2	Included Foundational WTA Literature by Citation Count ..... 43
3	Ratio-Based Assignment Scores Computation ..... 55
4	Slope-Based Assignment Scores Computation ..... 57
5	Test Problem Sizes ..... 59
6	Comparison of QP Heuristic and Construction Heuristic Over 20 Instances per Problem Size ..... 61
7	Average Computational Effort for Multiple Denials in the ED Metaheuristic ..... 64
8	Improvements for 1 Denial With Different Threshold Percentages for the Slope-based ED Metaheuristic ..... 64
9	Improvements for 2 Denials With Different Threshold Percentages for the Slope-based ED Metaheuristic ..... 65
10	Relative Optimality Gaps with BARON (%) ..... 67
11	Number of Solutions (out of 20 Instances per Problem Size) That Each Heuristic Solution is Superior to BARON ..... 68
12	Average Relative Gaps for Selected Heuristics with Respect to the Best Solution Available (%) ..... 69
13	Tested Problem Instance Sizes ..... 80
14	Average Solution Times and Hypothesis Tests ..... 81
15	<b>P1</b> Suboptimal Solutions ..... 83
16	Example of Greedy Hungarian Heuristic ..... 91
17	Example using Hungarian Algorithm ..... 92
18	Tested Problem Sizes ..... 93

Table		Page
19	Average Gap with BARON .....	94
20	Average Optimality Gap with BARON Using Transformation .....	95
21	Average Computational Time Requirements .....	96
22	Small Experimental Design .....	111
23	Solutions within n% Optimal (%) .....	112
24	Large Experimental Design .....	114
25	Experimental Scenarios .....	129
26	Single Stage Results.....	130
27	Two Stage Results .....	132

# Real-Time Heuristics and Metaheuristics for Static and Dynamic Weapon Target Assignments

## I. Introduction

This dissertation focuses on the weapon target assignment (WTA) problem. Specifically, it develops models by which air defense problems are framed and proposes new heuristic algorithms to find solutions in “real-time” to these models. We define a heuristic algorithm “real-time” if it is capable of finding a solution within the time it takes a missile to launch and reach its point of impact.

The WTA can be categorized as static (SWTA) or dynamic (DWTa). The SWTA models an air defense scenario wherein a known number of incoming missiles is observed and a given number of interceptors is available for a single engagement. The results of the assignments of interceptors to missiles is not explored, nor is the time frame in which the engagements are to occur. That is, if five incoming missiles are engaged by twenty-five interceptors, and each interceptor engages each of the five missiles, the solution would not explore how such a firing sequence would occur or whether any of the missiles would survive the engagements.

By contrast, the DWTa considers the outcomes of the SWTA as a subsequent problem. The “Shoot-Look-Shoot” variant requires the determination of which missiles, if any, survive an engagement and allows for a re-engagement. The “Two Stage” variant has a first stage which is identical to the SWTA and a second stage in which a number of incoming missiles is known only to a probability distribution. In each of these variants, the solution must consider how many of the interceptors to preserve for successive engagements or successive stages.

A second categorization which defines the WTA is the number of different types of interceptors. In the homogeneous problem, each interceptor is of the same type, thus the probability with which an interceptor will successfully destroy an incoming missile, or probability of kill, is the same for each interceptor. Alternatively, in the heterogeneous problem, there are different types of interceptors, each of which has different probabilities of kill, making the problem more complex. Indeed, the heterogeneous SWTA is an NP-Hard problem (Lloyd & Witsenhausen, 1986).

In this dissertation, the one and two stage heterogeneous WTAs are considered wherein each of the chapters is an independent scholarly article. Chapter 2 gives a contemporary survey of the literature on the WTA, focusing on models, optimal algorithms, and heuristic algorithms for the SWTA and DWTAs. In Chapter 3, a metaheuristic is presented that is capable of finding real-time solutions for the SWTA to some of the largest problems within the literature. A systemic problem identified in the use of the commercial solver, BARON, is explored in Chapter 4. A logarithmic transformation and a constraint with an instance-specific parameter are developed and used to improve the performance of BARON when solving the SWTA. A new heuristic is developed in Chapter 5, improving upon the work in Chapter 3 in solution quality and required computational effort. The model used for the SWTA in chapters 3, 4, and 5, originally presented by Manne (1958), is well known and extensively used within the literature. In Chapter 6, a model for the homogeneous two stage DWTAs developed by Murphey (2000) is used for the two stage heterogeneous DWTAs, which is solved therein. In Chapter 7, the continuous missile flight paths are modeled with Bézier curves and a solution technique for a DWTAs that solves the problem within a realistic simulation is developed.

In Chapter 2, the literature of the WTA since its inception in 1958 (Manne, 1958) is reviewed. Some of the different models for the SWTA and DWTAs are defined,



illustrating different parameters or alterations in the assumptions of the models. The survey proceeds to review the optimal algorithms for both the SWTA and DWTA. As previously stated, the heterogeneous SWTA is NP-Hard, thus the optimal algorithms generally apply to one of three models: the homogeneous WTA (for which an optimal solution is accessible in polynomial time), a linear transformation of the heterogeneous WTA, or a heterogeneous WTA with very few interceptors and missiles. A review of some of the heuristic and metaheuristic algorithms used on the WTA follows. Some of these, such as the genetic algorithm, have been implemented many times in the literature and some of these more well-cited works are explored. This chapter concludes by examining some of the recent developments within the literature, exploring some of the alternative applications of WTA research, and defining a method by which the high volume of literature is truncated to allow for a more comprehensive review.

Chapter 3 develops and presents a metaheuristic which improves upon previous research in which a heuristic that implements the solution to the quiz problem (i.e., the QP Heuristic) (Kline, 2017) was developed. This metaheuristic is called the Eminent Domain (ED) Metaheuristic since it finds improved solutions by denying a subset of the heuristic solution and resolving using the QP Heuristic. This denial process allows for the possibility of improved solutions by preventing those solutions of which a single assignment's selection, which is made by a greedy selection criterion, prevents subsequent assignments which may be of a superior solution. We compare the results of the ED to those of a heuristic developed by Ahuja et al. (2007), a known benchmark for the SWTA.

In Chapter 4, a logarithmic transformation to the SWTA formula is introduced with which the commercial solver BARON can more effectively find the optimal solution when possible and a lower bound when BARON is unable to determine an

optimal solution. However, this transformation is computationally expensive and a constraint with an instance-specific parameter is introduced to reduce the feasible region of the problem, thus improving the efficiency of BARON in solving this transformed problem.

Chapter 5 improves upon the QP subroutine by developing a new heuristic, which has similarities to the Hungarian Algorithm, called the Greedy Hungarian-like Heuristic (GH). This heuristic makes assignments by examining the best available assignments for each weapon and for each target and selecting those assignments that are among the best in both. The results of the GH are compared with the QP and ED, and the GH is used as a subroutine for the ED in further comparison.

In Chapter 6, the heterogeneous two stage DWTa is considered. Extending the work by Ahner & Parson (2015), this chapter uses the Concave Adaptive Value Estimation (CAVE) Algorithm, with the GH as a subroutine, to compute the number of interceptors to preserve for the second stage while using the GH to solve each stage given the solution of the CAVE Algorithm. This dynamic programming technique is used to approximate the value of each interceptor type in the second stage through the use of a simulation. The solutions of the CAVE Algorithm are compared to those of a baseline policy and an optimal policy for small problem instances. Additional comparative tests for the CAVE Algorithm and the baseline policies are conducted on larger problem instances for which finding the optimal policy is intractable.

Chapter 7 extends the work of Leboucher et al. (2013), who modeled the flight paths of incoming missiles in a one stage problem with two dimensional Bézier curves, requiring the determination of a firing sequence. This extension uses three dimensional Bézier curves so as to model the angle of approach of each missile and determine a firing sequence which is limited by the time required for each engagement and the windows of opportunity for engagement with each interceptor. Further, it determines

the outcome of each engagement and allows for subsequent engagements in a Shoot-Look-Shoot framework. This model is extended to two stages, thus modeling and solving a two stage Shoot-Look-Shoot problem. A simulation is used to test the efficacy of the solution technique, which initiates at the launch of incoming missiles and requires the solution of an engagement plan, as well as the reallocation of interceptors as battle damage is assessed, in real-time. The results of the solution technique are compared to those of a realistic Shoot-Shoot-Look policy, which engages a missile with two interceptors as soon as a missile is within the effective range of an interceptor.

The contributions of this dissertation are identified as follows. First, a real-time metaheuristic is developed, using the QP Heuristic as a subroutine, that is capable of efficiently finding near optimal solutions to the largest problems of the SWTA found in the literature and that dominates known solution techniques (Chapter 3). A logarithmic transformation with tight constraints is developed for the SWTA to improve the efficiency and quality of the solution when utilizing the commercial solver, BARON (Chapter 4). A new real-time heuristic algorithm is developed which can find quality solutions to the SWTA with greater speed than known heuristic algorithms and is implemented within a metaheuristic framework, finding the best real-time solutions to the SWTA, in terms of solution quality, in less time than required by the metaheuristic with the QP Heuristic subroutine (Chapter 5). A solution technique capable of solving the two stage heterogeneous DWTa is developed and presented for the first time in the literature, extending the work of Ahner & Parson (2015) (Chapter 6). A model for the DWTa is developed which considers the continuous flight path of incoming missiles and time parameters that restrict windows of fire for interceptors (Chapter 7). Further, a solution technique capable of solving the two stage Shoot-Look-Shoot DWTa is developed and its efficacy is tested within simulations (Chapter 7). Lastly, a contemporary survey of the literature of the WTA that highlights the

foundational contributions along with major contributions as measured by citation rate is presented (Chapter 2).

## II. Literature Review

### 2.1 Introduction

Projectile weapons have been a consistent threat of hostilities throughout history. Military advantage has always been aided by the capacity to inflict damage from a distance. In the 20<sup>th</sup> century, missile technology advanced to the point that an adversary had the potential to attack a protected asset from great distances. To neutralize this stand off threat, the concept of air defense evolved. However, as the ability to reduce a missile threat increased, so too did the quantity and quality of missiles available, and research into the effective allocation of air defense resources emerged.

Originally introduced into the field of operations research by Manne (1958), the Weapon Target Assignment (WTA) Problem, or Missile Allocation Problem (MAP) as it is sometimes known, seeks to assign available interceptors to incoming missiles so as to minimize the probability of a missile destroying a protected asset. While much of the literature on the WTA focuses on the defensive perspective, some have considered the offensive perspective (Sikanen, 2008), wherein the objective is to maximize the probability of destroying enemy protected assets.

There are two distinct categories of the WTA: the Static WTA (SWTA) and the Dynamic WTA (DWTa). Originally modeled by Manne (1958), the SWTA defines a scenario wherein a known number of incoming missiles (targets) are observed and a finite number of interceptors (weapons), with known probabilities of successfully destroying the targets (probabilities of kill), are available for a single exchange. The solution to the SWTA informs the defense on how many of each weapon type to shoot at each target. In the SWTA, no subsequent engagements are considered since time is not a dimension considered in the problem.

By contrast, the DWTa includes time as a dimension. Variants of the DWTa include the two stage DWTa and the shoot-look-shoot DWTa. The two stage DWTa replicates the SWTa in its first stage, but includes a second stage wherein a number of targets of various types are known only to a probability distribution. In this variant, the solution to the DWTa informs the defense on how to allocate the weapons in the first stage and how many to reserve for the second stage in order to minimize the probability of destruction. The shoot-look-shoot variant also replicates the SWTa, however it enables the defense to observe which targets may have survived the engagement (leakers) and allows for a subsequent engagement opportunity. The solution to this variant similarly informs the defense on how to allocate the weapons and how many weapons to reserve to reengage any leakers.

The WTa has been solved to optimality with exact algorithms. However, as Lloyd & Witsenhausen (1986) showed that the WTa is NP-Complete, the majority of solution techniques seek to find near optimal solutions in real-time, or “fast enough to provide an engagement solution before the oncoming targets reached their goals” (Leboucher et al., 2013). These real-time solution techniques are products of heuristic algorithms or are solved using exact algorithms applied to transformations of the formulation.

The rest of this paper proceeds as follows. In §2.2, we review the various formulations for both the SWTa and DWTa. We examine the basic formulations of each and explore the transformations which have been implemented. We also review novel formulations which have sought to model and solve the problem in unique settings. In §2.3, we review the exact algorithms that have been used to solve the SWTa and DWTa. Some of these algorithms provide optimal solutions to the original formulations whereas others refer to the transformed formulations identified in §2.2. In §2.4, we review the heuristic and metaheuristic solution techniques for the SWTa

and DWTa. In §2.5, we discuss the state of the WTA and present a metric with which we focused this examination of the literature.

## 2.2 Formulations

There have been many different formulations of the WTA. Early literature sought to transform the nonlinear formulation from Manne (1958) due to the computational limitations with nonlinear programming. As computational power increased, transformations which were better suited to global optimization tools emerged. Burr et al. (1985) introduced the DWTa which captured the value of subsequent engagements. Similar to the SWTA, variations to the original DWTa occur throughout the literature.

Herein, we examine some of the formulations for both the SWTA and DWTa. For purposes of clarity in both formulation and in presentation, we map the formulations presented by their authors into the terms of the formulation developed by Manne (1958). Namely, variables that are shared between multiple formulations are defined as follows:

$p_{ij}$  : the probability weapon  $i$  destroys target  $j$   
 $q_{ij}$  : the probability weapon  $i$  fails to destroys target  $j$   
 $V_j$  : the destructive value of target  $j$   
 $x_{ij}$  : the number of weapons of type  $i$  assigned to target  $j$   
 $K$  : the number of protected assets  
 $a_k$  : the value of asset  $k$   
 $n$  : the number of targets  
 $m$  : the number of weapon types  
 $w_i$  : the number of weapons of type  $i$   
 $c_{ij}$  : a cost parameter for assigning a weapon of type  $i$  to target  $j$   
 $\mathcal{F}$  : the set of feasible assignments  
 $\gamma_{jk}$  : the probability target  $j$  destroys asset  $k$   
 $s_j$  : the maximum number of weapons that can be assigned to target  $j$   
 $t$  : the number of stages

### **SWTA Formulations.**

The original formulation as defined by Manne (1958) considers a scenario where a defender has  $w_i$  of  $i = 1, \dots, m$  weapon types with which to defend against  $j = 1, \dots, n$  targets. Each weapon type  $i$  has a probability  $p_{ij}$  of killing target  $j$  and each target  $j$  has a destructive value  $V_j$ . With decision variables  $x_{ij}$  indicating the number



of weapons of type  $i$  to assign to target  $j$ , the SWTA is formulated:

$$\begin{aligned}
& \min \sum_{j=1}^n V_j \prod_{i=1}^m (1 - p_{ij})^{x_{ij}} \\
& \text{s.t. } \sum_{j=1}^n x_{ij} \leq w_i, \quad \text{for } i = 1, \dots, m \\
& x_{ij} \in \mathbb{Z}_+, \quad \text{for } i = 1, \dots, m, j = 1, \dots, n.
\end{aligned}$$

However, it is common to write the formulation in terms of the probability of survival  $q_{ij} = 1 - p_{ij}$

$$\begin{aligned}
\mathbf{S1} \quad & \min \sum_{j=1}^n V_j \prod_{i=1}^m q_{ij}^{x_{ij}} \\
& \text{s.t. } \sum_{j=1}^n x_{ij} \leq w_i, \quad \text{for } i = 1, \dots, m \\
& x_{ij} \in \mathbb{Z}_+, \quad \text{for } i = 1, \dots, m, j = 1, \dots, n.
\end{aligned}$$

The nonlinear objective function in **S1** seeks those assignments to minimize the expected value of survival. The assignments are integer and the total number of weapon  $i$  cannot exceed the number of weapons on hand,  $w_i$ . This formulation is used frequently, (e.g., (Ahuja et al., 2007), (Lemus & David, 1963), (Lee et al., 2002), (denBroeder et al., 1959)) and is often the initial formulation used when implementing a transformation.

A simpler version of **S1** is given by denBroeder et al. (1959), who assumes that all weapons have the same probability of kill for target  $j$ ,  $p_{ij} = p_j \forall i = 1, \dots, m$ . His formulation differs from **S1** in the objective, which is

$$\mathbf{S2} \quad \min \sum_{j=1}^n V_j q_j^{x_j}$$

This formulation simplifies **S1** and is easily optimized by a greedy assignment technique. However, its assumption of homogeneity greatly reduces the applicability of the formulation.

Kwon et al. (1999) utilize a similar model to **S1** but reformulate the problem into an integer program with a linear objective function and nonlinear constraints. They use a negative cost parameter,  $c_{ij}$ , for assigning weapon  $i$  to target  $j$  which they seek to minimize as follows:

$$\begin{aligned}
& \min \sum_{(i,j) \in A} c_{ij} x_{ij} \\
& \text{s.t.} \quad \sum_{\{j=1, \dots, n \mid (i,j) \in A\}} x_{ij} \leq w_i \quad \text{for } i = 1, \dots, m \\
& \quad 1 - \prod_{\{i=1, \dots, m \mid (i,j) \in A\}} (1 - p_{ij})^{x_{ij}} \geq d_j \quad \text{for } j = 1, \dots, n \\
& \quad x_{ij} \leq u_{ij} \quad \forall (i, j) \in \mathcal{F}, \\
& \quad x_{ij} \geq 0 \quad \forall (i, j) \in \mathcal{F}
\end{aligned}$$

where  $d_j$  is the minimum desired probability of kill for target  $j$ ,  $u_{ij}$  is an upper bound on the number of weapons  $i$  that can be assigned to target  $j$ , and  $\mathcal{F}$  is the set of all feasible assignments. Kwon et al. (1999) then multiply a large number  $\theta$  to a logarithmic transformation of the nonlinear constraint and round down to the largest integer contained in order to generate the following linear approximation,

where  $a_{ij} = \lfloor -\theta \ln(1 - p_{ij}) \rfloor > 0$  and  $b_j = \lfloor -\theta \ln(1 - d_j) \rfloor > 0$

$$\begin{aligned}
\mathbf{S3} \quad & \min \sum_{(i,j) \in A} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{\{j=1, \dots, n \mid (i,j) \in A\}} x_{ij} \leq w_i \quad \text{for } i = 1, \dots, m \\
& \sum_{\{i=1, \dots, m \mid (i,j) \in A\}} a_{ij} x_{ij} \geq b_j \quad \text{for } j = 1, \dots, n \\
& x_{ij} \leq u_{ij} \quad \forall (i, j) \in A, \\
& x_{ij} \geq 0 \quad \forall (i, j) \in A
\end{aligned}$$

This formulation is linear and is computationally simpler than **S1** and a solution is more easily attained. Because the formulation is an approximation, however, its solution is not guaranteed to be optimal for **S1**.

A different transformation to **S1** is put forth by Ahuja et al. (2007) by applying a logarithmic transformation to the objective. Letting  $d_{ij} = -\ln(q_{ij})$ , their formulation becomes

$$\begin{aligned}
\mathbf{S4} \quad & \min \sum_{j=1}^n V_j 2^{-y_j} \\
\text{s.t.} \quad & \sum_{j=1}^n x_{ij} \leq w_i \quad \text{for } i = 1, \dots, m \\
& \sum_{i=1}^m d_{ij} x_{ij} = y_j \quad \text{for } j = 1, \dots, n \\
& x_{ij} \in \mathbb{Z}_+ \quad \text{for } i = 1, \dots, m, j = 1, \dots, n \\
& y_j \geq 0 \quad \text{for } j = 1, \dots, n
\end{aligned}$$

With this transformation, Ahuja et al. (2007) have an objective which is the sum of separable convex functions. They utilize this transformation to model the SWTA as

a network flow problem, which is addressed later in section §2.1. Further, as is shown by Kline et al. (2017), utilizing **S4** within a commercial global optimization solver such as BARON is more reliable than when utilizing **S1**, which has roughly a 21% false optimality rate.

Others simplify the problem by limiting the number of weapons of each type to  $w_i = 1$ , making the problem a binary program. Li et al. (2009) propose the objective function

$$\mathbf{S5} \quad \min \sum_{j=1}^n V_j \prod_{i=1}^m (1 - p_{ij}x_{ij}),$$

with an added constraint which limits  $x_{ij}$  to a binary decision variable. **S1** can be transformed to **S5** by setting the number of weapon types to the total number of weapons. That is, if  $w_i = 3$  and  $m = 5$ , the problem could be transformed for **S5** by setting  $w_i = 1$  and  $m = 15$ . This increases the number of decision variables of the problem, though the transformation to a binary program allows for more efficient solution techniques.

A more simplified formulation is put forth by Rosenberger et al. (2005), who model the SWTA as a knapsack problem. They define a positive cost parameter  $c_j$ , which is earned when assignment  $j$  is selected. Their model assumes that no two weapons

can be assigned to the same target and is

$$\begin{aligned}
\mathbf{S6} \quad & \max \sum_{j \in J} c_j x_j \\
\text{s.t.} \quad & \sum_{j \in S_i} x_j \leq 1 \quad i = 1, \dots, m \\
& \sum_{j \in T_j} x_j \leq 1 \quad j = 1, \dots, n \\
& x_j = \begin{cases} 1 & \text{assignment } j \text{ is selected} \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

In the first constraint, the set  $S_i$  is the subset of all feasible assignments of which weapon  $i$  is assigned. Similarly, the set  $T_j$  in the second constraint is the subset of all feasible assignments which assigns a weapon to target  $j$ . While simpler than **S1**, this formulation, like **S2**, carries more assumptions which limit its ability to model and solve complex missile defense problems.

Malcolm (2004) proposed a formulation with the same binary decision variables in which the objective is similar in structure to **S5**. He shows that, when weapon assignments are restricted to exactly one target and  $m = n$ , the objective can be written as

$$\begin{aligned}
\text{S7} \quad & \min \quad - \sum_{j=1}^n V_j \left( \sum_{i=1}^m x_{ij} p_{ij} \right) \\
& \text{s.t.} \quad \sum_{j=1}^n x_{ij} = 1 \quad \text{for } i = 1, \dots, m \\
& \quad \sum_{i=1}^m x_{ij} = 1 \quad \text{for } j = 1, \dots, n \\
& \quad x_{ij} = \begin{cases} 1 & \text{weapon } j \text{ is assigned to target } i \\ 0 & \text{otherwise} \end{cases} \quad \text{for } i = 1, \dots, m, j = 1, \dots, n
\end{aligned}$$

This allows for solution techniques which exploit the special structure of the formulation, but is only of use under certain rigid situations.

Two additional variants to the SWTA formulations are also in the literature. One, defined by Shang et al. (2007), considers the value,  $a_k$ , of a protected asset  $k = 1, \dots, K$  and the probability with which a target  $j$  will destroy this asset  $\gamma_{jk}$ . Given the probability that weapon  $i$  will destroy target  $j$ ,  $p_{ij}$ , they formulate

$$\begin{aligned}
\text{S8} \quad & \min \quad \sum_{k=1}^K a_k \prod_{j=1}^{n_k} \left[ \gamma_{jk} \prod_{i=1}^W (1 - p_{ij} x_{ij}) \right] \\
& \text{s.t.} \quad \sum_{k=1}^K n_k = n \\
& \quad \sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, m \\
& \quad x_{ij} \in \{0, 1\}, \quad i = 1, \dots, m, j = 1, \dots, n
\end{aligned}$$

where each protected asset has incoming targets  $1, \dots, n_k$  and there are a total of  $n$  targets. This formulation considers the importance of different protected assets,

which is relevant in a missile defense problem, but it adds complexity to the problem.

Karasakal (2008) does not consider a target value but rather treats each target to be of identical destructive capacity. He defines the set of feasible solutions as  $\mathcal{F}$ , limits the number of weapons that can be assigned to target  $j$  as  $s_j$ , and defines a formulation

$$\begin{aligned}
\text{S9} \quad & \max \quad \prod_{j=1, \dots, n} \left[ 1 - \prod_{\{i=1, \dots, m \mid (i,j) \in \mathcal{F}\}} (1 - p_{ij})^{x_{ij}} \right] \\
& \text{s.t.} \quad \sum_{\{j=1, \dots, n \mid (i,j) \in \mathcal{F}\}} x_{ij} \leq w_i \quad \text{for } i = 1, \dots, m \\
& \quad \sum_{\{i=1, \dots, m \mid (i,j) \in \mathcal{F}\}} x_{ij} \leq s_j \quad \text{for } j = 1, \dots, n \\
& \quad 0 \leq x_{ij} \leq u_{ij}, \quad \forall i, j \in \mathcal{F} \text{ and } x_{ij} \text{ is integer}
\end{aligned}$$

This formulation treats all protected assets and targets as having equal value and simply seeks to maximize the expected destruction to incoming targets.

### **DWTA Formulations.**

#### **Shoot-Look-Shoot.**

There are two variants of the DWTA, each of which have unique formulations. The first variant is the shoot-look-shoot scenario, wherein weapons are assigned to targets in a first engagement and a subsequent engagement allows assigning remaining weapons to any surviving targets. This problem was discussed by Eckler & Burr (1972), who do not define a model but define the probability that  $n$  targets are destroyed over  $t$  stages, which is equivalent to the probability that at least  $n$  weapons

do not fail over  $t$  stages, as

$$P(t) = \sum_{i=n}^t \binom{t}{i} (1-p)^{t-i} p^i, \quad (1)$$

for a problem wherein all probabilities of kill are the same and all targets are of the same value. Eckler & Burr (1972) identify that the most desired strategy to the  $t$  stage problem under this assumption will be equivalent to finding the number of weapons to use in each stage which minimize the number of stages necessary to achieve some acceptable value of  $P(t)$ .

Soland (1987) provides a model for the Eckler & Burr (1972) scenario with the assumption that all weapons have the same probabilities of kill and all targets have the same value. Given a nondecreasing function  $g(n_q)$  which defines the expected fraction of targets destroyed, where  $n_q$  is the number of unintercepted targets, the state of the system  $S(n_q, d, t)$  defines the fraction of targets destroyed given  $n_q$  targets,  $d$  weapons, and  $t$  remaining engagements. The state space is bound by

$$S(0, d, t) = 0, \quad \text{for } d = 0, 1, \dots, D, \quad t = 1, \dots, T,$$

$$S(n_q, d, 0) = g(n_q), \quad \text{for } n_q = 0, 1, \dots, n, \quad d = 0, 1, \dots, D.$$

He defines the transition probability that  $j$  targets survive having used  $i$  weapons as  $P(j|n_q, i, d, t)$ , where

$$P(j|n_q, i, d, t) = \binom{n_q}{j} q_t^{jI} (1 - q_t^I)^{n_q - j}.$$

$I$  defines the spread of weapons to targets, or  $I = \frac{i}{n_q}$  in the case that  $\frac{i}{n_q}$  is integer. If it is not integer, then  $n_q + n_q \left\lfloor \frac{i}{n_q} \right\rfloor - i$  of the targets receive  $\left\lfloor \frac{i}{n_q} \right\rfloor$  weapons and the remaining  $i - n_q \left\lfloor \frac{i}{n_q} \right\rfloor$  targets receive  $\left\lceil \frac{i}{n_q} \right\rceil$  weapons. His model seeks to minimize the



number of weapons required to ensure that the expected number of surviving targets is less than some “nondecreasing maximum damage function  $f$ ” (Soland, 1987)

$$\begin{aligned} \mathbf{D1} \quad & \min D \\ \text{s.t.} \quad & S(n_q, D, T) \leq f(n_q), \quad n_q = 1, \dots, n. \end{aligned}$$

Hosein & Athans (1989) provide a different model than Soland (1987), but with the same underlying assumptions. They define  $n_k(t)$  as the number of targets aimed at protected asset  $k$  at stage  $t$ , or the number of surviving targets after  $t-1$  engagements. They compute the probability that the number of targets surviving into the second stage is  $j^{(2)}$  given the assignment in the first stage is  $x^{(1)}$ , for all  $i = 0, 1, \dots, m$  and  $j = 0, 1, \dots, n(1)$  as  $P(n(2) = j^{(2)} | x^{(1)})$ . Defining  $J_s^*(n(2), m_2)$  as the optimal solution to the second stage with  $m_2$  weapons available, they define the formulation

$$\begin{aligned} \mathbf{D2} \quad & \max_{x^{(1)} \in \mathbb{Z}_+^K} J_d = \mathbb{E}_{n(2)} [J_s^*(n(2), m_2)] \\ \text{s.t.} \quad & |x^{(1)}| + m_2 = m. \end{aligned}$$

The objective is the expected value of the optimal solution in the second stage, thus the optimal solution to the problem is to find the number of weapons to use in the first stage,  $m_1$ , and assign them to the appropriate targets,  $x^{(1)}$ , in such a way that the second stage can be solved to optimality given the number of surviving targets and the number of unused weapons,  $m_2$ .

A different approach, which considers the available windows in which targets can be engaged, is proposed by Leboucher et al. (2013). He computes random paths of randomly located targets using Bézier curves, which allow for the calculation of the

time to impact for each target and the earliest point at which each weapon can engage the target. For each weapon-target pairing, he computes:

$$\begin{aligned} f_1(E_{i/j}) &= EFF_{i/j}, \quad (i \in I), (j \in J) \\ f_2(E_{i/j}) &= LFF_{i/j} - EFF_{i/j}, \quad (i \in I), (j \in J) \\ f_3(E_{i/j}) &= d(P_{j_{out}}, P_{i_0}) \end{aligned}$$

where  $EFF_{i/j}$  is the earliest feasible fire time for weapon  $i$  to target  $j$ ,  $LFF_{i/j}$  is the latest feasible fire time for weapon  $i$  to target  $j$ , and  $d(P_{j_{out}}, P_{i_0})$  is the Euclidean distance that the weapon,  $i_0$ , must fly over the protected area to intercept target  $j$ .

Using these three parameters for each pairing, Leboucher et al. (2013) creates a cost matrix for all of the possible assignments

$$H = \begin{bmatrix} E_{1/1} & E_{2,1} & \cdots & E_{|I|/1} \\ E_{1/2} & E_{2,2} & \cdots & E_{|I|/2} \\ \vdots & \vdots & \ddots & \vdots \\ E_{1/|J|} & E_{2,|J|} & \cdots & E_{|I|/|J|} \end{bmatrix},$$

in which the cost of an assignment  $H(E_{i/j})$  is

$$H(E_{i/j}) = \alpha_1 f_1(E_{i/j}) + \alpha_2 f_2(E_{i/j}) + \alpha_3 f_3(E_{i/j}),$$

where  $\alpha_1 + \alpha_2 + \alpha_3 = 1$  and  $(\alpha_1, \alpha_2, \alpha_3) \in [0, 1]^3$ . He presents a formulation to

determine the assignments

$$\begin{aligned}
\mathbf{D3a} \quad & \min \sum_{i=1}^m \sum_{j=1}^n H(E_{i/j}) x_{ij} \\
\text{s.t.} \quad & \sum_{i=1}^m \sum_{j=1}^n x_{ij} \leq m
\end{aligned}$$

and defines, for each possible firing sequence (FS), three parameters. First, given the time at which target  $j$  is engaged,  $FT_j$ , the parameter measuring the firing time is  $f_4(FS) = \sum_{j=1}^T FT_j$ . Second, a parameter identifying any constraint violation is  $f_5(FS) = \sum_{i=1}^m c_i$ , where  $c_i$  is 1 if the assignment of weapon  $i$  violates a constraint and 0 otherwise. Lastly, the parameter representing idle time of the system, given the time at which weapon  $i$  is fired,  $FT_i$ , is  $f_6(FS) \sum_{i=1}^{m-1} (FT_{i+1} - FT_i)$ . Leboucher et al. (2013) present a formulation whose solution gives the optimal firing sequence of the assignment solution

$$\begin{aligned}
\mathbf{D3b} \quad & \min F(FS) \\
\text{s.t.} \quad & F(FS) = \begin{cases} (f_4(FS) + 1) * f_6(FS) & f_5(FS) = 0 \\ \infty & f_5(FS) \neq 0 \end{cases}
\end{aligned}$$

The formulation presented by Leboucher et al. (2013) enables observation of surviving targets following an engagement which can be reengaged in a subsequent iteration and can be used to solve a shoot-look-shoot problem.

## 2-Stage.

The second variant of the DWTA is the 2 stage, or more generally the multi-stage, problem, which differs from the shoot-look-shoot in that it does not allow the

reacquisition of leakers. That is, in the shoot-look-shoot problem, a given number of targets are repeatedly engaged until all have been destroyed or a limit to the number of iterations is met. In the 2 stage problem the given number of targets is only engaged once before a subsequent stage occurs. In the second stage, the number and type of incoming targets is known only to a probability distribution.

Chang et al. (1987) model the  $T$  stage WTA by considering the value of each stage as defined by the formulation **S1** and taking the sum over the  $T$  stages.

$$\begin{aligned}
\mathbf{D4} \quad & \min_{x_{ij}(t)} \quad \mathbb{E} \left[ \sum_{t=1}^T \sum_{j \in \mathcal{A}_t} V_j(t) \prod_{i=1}^m (1 - p_{ij}(t))^{x_{ij}(t)} \right] \\
\text{s.t.} \quad & \mathcal{A}_{t+1} = (\mathcal{A}_t \cup \mathcal{L}_t) \cap \mathcal{K}'_t \\
& M_i(t) = M_i(t-1) - \sum_{j=1}^n x_{ij}(t-1) \quad i = 1, \dots, m \\
& \sum_{j \in \mathcal{A}_t} x_{ij}(t) \leq M_i \quad i = 1, \dots, m \\
& x_{ij}(t) \in \mathbb{Z}_+ \quad i = 1, \dots, m, \quad j = 1, \dots, n
\end{aligned}$$

where  $\mathcal{A}_t$  is the set of targets in stage  $t$ ,  $M_i(t)$  is the number of interceptors of type  $i$  at stage  $t$ ,  $\mathcal{L}_t$  is the set of new targets observed in stage  $t$  and  $\mathcal{K}_t$  is the set of targets killed in stage  $t$ .

Burr et al. (1985) present a formulation for a multi-stage problem in which all weapons have the same probability of kill for all equally valued targets. That is, a known number of targets arrives in the first stage and a number arrive in each of the number of subsequent stages, both of which are known only to a probability distribution. Given the attack strategy  $a(k)$ , which identifies the number of targets aimed at asset  $k$ , and a defense strategy  $d(k, j)$ , which defines how many weapons to

shoot at the target  $j$  threatening asset  $k$ , Burr et al. (1985) formulate the problem

$$\begin{aligned} \mathbf{D5} \quad & \min_{d(k,j)} \sum_{k=1}^K \sum_{j=1}^{a(k)} d(k,j) \\ \text{s.t.} \quad & V(d, a) \leq \frac{V}{n} \sum_{k=1}^K a(k), \text{ for all } a \end{aligned}$$

where  $V$  is the sum of all protected asset values,  $n$  is the total number of targets, and  $V(d, a)$  is the expected damage to the protected asset given  $a$  targets and deployment strategy  $d$

$$V(d, a) = \sum_{k=1}^K a_k \left( 1 - \prod_{j=1}^{a(k)} (1 - q^{d(k,j)}) \right).$$

This formulation seeks to ensure that the expected damage to the protected assets is less than the total value of all protected assets.

Murphey (2000) formulates the multi-stage problem by defining  $n(t)$  as the number of targets which arrive at time  $t = 1, \dots, T$  and  $c(t)$  as a nondecreasing function which represents a cost of waiting. With the assumption that all weapons have the same probability of kill for target  $j$ , his formulation is

$$\begin{aligned} \mathbf{D6} \quad & \min \sum_{t=1}^T c(t) \sum_{j=1}^{n(t)} V_j q_j^{x_j(t)} \\ \text{s.t.} \quad & \sum_{t=1}^T \sum_{j=1}^{n(t)} x_j(t) = m, \\ & V_j \in \mathbb{V} \in \mathbb{R}_+^n \quad j = 1, \dots, n(T) \\ & x^{(t)} \in \mathbb{Z}_+^{n(t)} \quad t = 1, \dots, T \end{aligned}$$

where  $m$  is the total number of weapons and  $\mathbb{V}$  is the set of all target values. This formulation is seeking to minimize the value of the assignments and the inclusion of a non-decreasing cost of waiting function will bias the solution to make earlier

assignments unless these assignments are to targets of a very small value relative to those of later assignments.

Xin et al. (2011) allow for different probabilities of kill for each weapon to each target and further allow for different probabilities of kill between stages. Their formulation for stages  $t = 1, \dots, T$  is

$$\begin{aligned}
\mathbf{D7} \quad & \min \quad \sum_{k=1}^{K(t)} V_k \prod_{j=1}^{n(t)} \left( 1 - \gamma_{jk} \prod_{h=t}^T \prod_{i=1}^{m(t)} (1 - p_{ij}(h))^{x_{ij}(h)} \right) \\
& \text{s.t.} \quad \sum_{j=1}^{n(t)} x_{ij}(t) \leq n_i \quad \text{for } i = 1, \dots, m, \quad t = 1, \dots, T \\
& \quad \sum_{i=1}^{m(t)} x_{ij}(t) \leq s_j \quad \text{for } j = 1, \dots, n, \quad t = 1, \dots, T \\
& \quad \sum_{t=1}^T \sum_{j=1}^{n(t)} x_{ij}(t) \leq w_i \quad \text{for } i = 1, \dots, m \\
& \quad x_{ij}(t) \leq f_{ij}(t) \quad \text{for } i = 1, \dots, m, \quad j = 1, \dots, n, \quad t = 1, \dots, T
\end{aligned}$$

where

$K(t)$  : number of existing assets at time  $t$

$n(t)$  : number of existing targets at time  $t$

$m(t)$  : number of available weapons at time  $t$

$a_k$  : the value of asset  $k$

$\gamma_{jk}$  : the probability target  $j$  destroys asset  $k$

$p_{ij}(t)$  : the probability weapon  $i$  destroys target  $j$  at time  $t$

$n_i$  : maximum number of targets weapon  $i$  can shoot at each stage

$s_j$  : maximum number of weapons that can be assigned to target  $j$  at each stage

$w_i$  : total number of weapons of type  $i$

$f_{ij}(t)$  : 1 if weapon  $i$  can be assigned to target  $j$ , 0 otherwise.

This is one of the more complex and realistic models that can be found within the literature. It allows for expansion into a shoot-look-shoot problem and considers many parameters which are relevant to modeling missile defense. However, as the complexity is higher than other formulations, finding solutions is more computationally expensive than for simpler formulations.

A model proposed by Khosla (2001) considers the required time for weapon system control and defines the following terms:

$n$  : Number of current threats

$m$  : Number of current weapon systems

$T$  : Total number of time points in time interval

$TV(j)$  : Threat value of threat  $j$

$OW(i, j)$  : Option weight of weapon system  $i$  for threat  $j$

$LB(i, j)$  : Begin launch time for weapon system  $i$  for threat  $j$

$LE(i, j)$  : End launch time for weapon system  $i$  for threat  $j$

$GT(i, j)$  : Guidance time interval for interceptor for weapon system  $i$  to engage threat  $j$

$IR(i)$  : Inventory resource of weapon system  $i$  (number of interceptors)

$GR(i)$  : Guidance resource capacity of weapon  $i$

where  $GT(i, j)$  defines the amount of time the guidance system must be allocated to weapon  $i$  in targeting target  $j$  and  $GR(i)$  is the number of guidance systems available. The option weight  $OW(i, j)$  serves to add a benefit to preferred pairings; a bias for weapon  $i$  to be assigned to target  $j$ .

Khosla (2001) defines a mixed integer program with only a few of the considerations discussed thus far, proposing that expanded models including additional time constraints such as reload time. Using a decision variable  $L(i, j, t) = 1$  if  $t$  denotes the launch time for an interceptor from weapon system  $i$  to engage threat  $j$  and 0



otherwise, he models this problem, with a weight factor  $\alpha \in [0, 1]$ , as

$$\begin{aligned}
\mathbf{D8} \quad & \max \sum_{i=1}^m \sum_{j=1}^n \sum_{t=1}^T [\alpha TV(j) + (1 - \alpha) OW(i, j)] L(i, j, t) \\
\text{s.t.} \quad & \sum_{i=1}^m \sum_{t=1}^T L(i, j, t) \leq 1, \quad \text{for } j = 1, \dots, n \\
& \sum_{j=1}^n \sum_{t=1}^T L(i, j, t) \leq IR(i), \quad \text{for } i = 1, \dots, m \\
& \text{If } L(i, j, t) = 1, \quad R(i, j, t') = 1, \quad \forall t \leq t' \leq t + GT(i, j) \\
& \sum_{j=1}^n \sum_{t=1}^T R(i, j, t) \leq GR(i), \quad \text{for } i = 1, \dots, m
\end{aligned}$$

In this model, Khosla (2001) ensures that only one weapon system is assigned to each target, the total number of interceptors does not exceed the inventory, and the guidance time required for the assignments does not exceed the number of guidance systems. He nests this model into a framework which updates the number of targets after each completed iteration and, as such, can be used in a 2 stage scenario where the number of targets in subsequent stages is stochastically determined.

This model simplifies some of the parameters and considerations posed in **D7**, but includes a discretized time step which provides the firing sequence inherent to, but previously not considered, missile defense. However, due to the exponential growth of the number of decision variables with the increase in resolution, Khosla (2001) identifies that even modest sized problems are very computationally expensive.

Ahner & Parson (2015) address the SWTA formulation, originally proposed by Murphey (2000), to compute the value of the first stage and includes in the objective the expected value of the second stage, with the maximum number of weapons,  $b < M$ ,

that can be used in any stage, as follows

$$\begin{aligned}
\mathbf{D9} \quad & \min_x \left\{ \sum_{j=1}^{n_1} V_j^{(1)} \left(1 - p_j^{(1)}\right)^{x_j^{(1)}} + \mathbb{E}_{\omega \in \Omega} \left[ Z_2 \left(x^{(2)}, \omega^j\right) \right] \right\} \\
& \text{s.t.} \quad x^{(1)} \leq b \\
& \quad x_j \in \mathbb{Z}_+, \text{ for } j = 1, \dots, n
\end{aligned}$$

where  $x_j^{(1)}$  defines the number of weapons fired at target  $j$  in the first stage and the total number of weapons fired in the first stage is  $x^{(1)} = \sum_{j=1}^{n_1} x_j^{(1)}$ . Further, the second stage,  $Z_2$ , is a function of the remaining weapons,  $x^{(2)} = M - x^{(1)}$ , and a random occurrence,  $\omega \in \Omega$  of the number of targets and the type of each target.

$$\begin{aligned}
Z_2 \left(x^{(2)}, \omega^j\right) &= \min_{x^{(2)}} \left\{ \sum_{j=1}^{n_2(\omega)} V_j^{(2)}(\omega) \left(1 - p_j^{(2)}\right)^{x_j^{(2)}} \right\} \\
& \text{s.t.} \quad \sum_{j=1}^{n_1} x_j^{(1)} + \sum_{j=1}^{n_2(\omega^j)} x_j^{(2)} = M \\
& \quad x^{(2)} \leq b \\
& \quad x_j^{(2)} \in \mathbb{Z}_+, \text{ for } j = 1, \dots, n
\end{aligned}$$

Unlike other formulations, this formulation accounts for the uncertainty of a second stage which must be considered when allocating available weapons to the first stage. However, it is a homogeneous model which assumes that all weapons have the same probability of kill for a target  $j$ .

Sikanen (2008) models the DWTA wherein all weapons have the same probability

of kill for all targets

$$\begin{aligned}
\text{D10} \quad & \max_{x_{i1}, \dots, x_{iT}} \sum_{t=1}^T \sum_{j=1}^{n(t)} p_j(t) \lambda_j^t V_j x_{jt} \\
\text{s.t.} \quad & \sum_{t=1}^T x_{jt} \leq 1 \quad \text{for } j = 1, \dots, n(t) \\
& \sum_{t=1}^T \sum_{j=1}^{n(t)} x_{jt} \leq m \\
& x_{jt} \in \{0, 1\} \quad \text{for } j = 1, \dots, n(t), \quad t = 1, \dots, T
\end{aligned}$$

where there are  $T$  stages,  $n(t)$  targets per stage, and  $m$  available weapons. Further, the value of target  $j$  is  $V_j$  and its time discount factor at stage  $t$  is  $\lambda_j^t$ . Given that  $x_{jt}$  is a binary decision variable, the expression is the sum over all stages of the sum over all assignments. The time discounting factor imposes a bias on earlier assignments. The product of the probability, target value, time discount factor, and decision variable result in either a value of 0 or the time discounted expected value of the assignment. Though a superior assignment may occur as the target is closer to the protected asset ( $p_j(t)$  is a function of time), the value of the target decreases since a miss will reduce the ability to reengage.

### 2.3 Exact Algorithms

There are few cases in the literature of exact algorithmic solutions to the WTA. The problem suffers due to its complexity as an NP-Complete problem (Lloyd & Witsenhausen, 1986) and, like routing problems, are simply hard to solve. For the SWTA, the number of possible permutations of assigning  $m$  weapons to  $n$  targets is  $n^m$ , assuming that all weapons must be assigned, which, as the number of weapons and targets increases, grows exponentially and searching all possible solutions quickly

becomes computationally intractable. Because the DWTA includes either a shoot-look-shoot or multiple stage (or both) framework, it further increases the number of permutations. The literature implementing exact solution techniques generally fall into one of two categories: small problems and problems wherein assumptions reduce the complexity.

### **SWTA.**

denBroeder et al. (1959) showed the first optimal solution technique in their Maximum Marginal Return (MMR) algorithm. Assuming that the probability of kill for any weapon to target  $j$  is the same, they showed that an optimal solution can be found by assigning  $x_{ij} = 1$  where  $\{i, j\} \in \arg \max(V_j p_{ij})$  and then updating  $V_j = V_j(1 - p_{ij})$  and  $p(i, \cdot) = p(\cdot, j) = 0$  and repeating the process until all weapons have been assigned. Further, when the probabilities of kill are the same for all weapons to all targets,  $p_{i_1 j_1} = p_{i_2 j_2}$ ,  $\forall i_1, i_2 \in I$ ,  $j_1, j_2 \in J$ , the optimal solution is found by dividing the weapons evenly across all targets (Hosein et al., 1988).

Malcolm (2004) developed and solved **S7**, where he defined his constraint coefficient matrix  $A$  as totally unimodular. This ensures that every vertex of the convex polytope that defines the feasible solution space is an integer solution. As such, he uses the Simplex Method to quickly find the optimal solution.

Smaller problems were solved through an exhaustive search algorithm for **S1** by Johansson & Falkman (2009). In comparing the objective function value of every feasible solution, they show that a problem with 9 weapons and 8 targets took 13 minutes to run to completion and that adding one additional target took 43.7 minutes to run to completion, which they present to illustrate the combinatorial explosion in run time as a function of problem size.

Several cases of using a branch and bound algorithm are found in the literature.

Rosenberger et al. (2005) solved **S6** for up to 8 weapons and 4 targets. Ahuja et al. (2007) implemented three lower bounding strategies to increase the efficiency of fathoming nodes: a generalized network flow solution, an MMR solution, and a minimum cost flow solution. Kline (2017) developed a branch and bound algorithm to solve **S1** and was able to find optimal solutions for up to 10 weapons and 10 targets. Beyond this, the size of the problem precluded convergence within 7 days of computation.

Karasakal (2008) utilizes linear integer programming techniques to find optimal solutions to two linear transformations of **S9**.

Bogdanowicz (2012) develops and utilizes an algorithm by which he searches through known effective weapon-target pairings to find an optimal solution. Utilizing the Joint Munition Effectiveness Manual (JMEM), he defines the desired minimal effect of any one pairing to reduce the number of sets through which he searches for an optimal set of pairings, given the number and type of weapons and targets.

## **DWTA.**

Burr et al. (1985) puts forth an optimal algorithm for **D4** given a scenario wherein one target per stage is observed and a defender will assign weapons for up to  $k - 1$  stages, after which he will surrender the protected asset. Given a maximum total expected damage of 1, the defender must limit the expected damage per stage to no greater than  $r = \frac{1}{k}$ , where each weapon has a probability of kill of  $p = 1 - q$ . His algorithm is to set the minimum number of weapons to ensure damage does not exceed  $r$  for the first stage as

$$d(1) = \left\lceil \frac{\ln(r)}{\ln(q)} \right\rceil$$

and all subsequent stages as

$$d(k') = \left\lceil \frac{\ln \left( 1 - \frac{1-rk'}{\prod_{i=1}^{k'-1} (1-q^{d(i)})} \right)}{\ln(q)} \right\rceil,$$

where  $1 < k' \leq k - 1$ .

Soland (1987) gives an optimal solution to the shoot-look-shoot model **D1** in which the number of weapons  $i$  to assign to the total number of targets  $a$  in the first stage is simply  $\lfloor \frac{i}{a} \rfloor$  and to preserve the remaining  $i - \lfloor \frac{i}{a} \rfloor$  weapons for the surviving targets. If the problem allows for more than two stages, he iteratively performs this allocation, utilizing the largest integer contained in the fraction of available weapons to surviving targets in the immediate stage and preserving the remaining weapons for the subsequent stage.

Hosein (1989) proves that the optimal solution to **D2** is to spread the number of weapons used for each stage  $t$ ,  $m_t$ , as evenly as possible, which is similar to Soland (1987). He therefore seeks to optimize over the decision variables  $m_t$  the minimum value of the final stage.

Ahner & Parson (2015) generate an optimal strategy for **D9** through the implementation of the Concave Adaptive Value Estimation (CAVE) algorithm with a modified MMR algorithm which they call the MMR Plus Algorithm. The CAVE algorithm estimates the value of second stage assignments by utilizing random realizations of the number of targets in the second stage and iteratively updating the subgradient of a concave value estimation, the CAVE function. Their MMR Plus algorithm assigns weapons to known targets in the first stage and, by comparing marginal returns of assignments to the CAVE function, indicates how many weapons to preserve for the second stage. Though the CAVE Algorithm is an approximation technique, Ahner & Parson (2015) prove the convergence to the optimal solution in

the DWTA wherein all weapons have the same probabilities of kill to target  $j$ .

Sikanen (2008) uses dynamic programming to solve **D10**. He uses a backwards induction process to recursively define the policy which will optimize the problem.

## 2.4 Heuristic Algorithms

Due to the computational complexity of the WTA, much of the literature focuses on heuristic algorithms which provide real time solutions rather than guaranteed optimal solutions. Many of these are of well known heuristic algorithms, such as the very large scale neighborhood (VLSN) search or the Genetic Algorithm (GA), but others are of new design, seeking to exploit the special structure of the WTA.

### **SWTA.**

The heuristic algorithms applied to the SWTA often fall into one of several groups. Herein, we will explore some of the varying approaches within these groups.

### **MMR.**

Kolitz (1988) implemented the MMR algorithm and, unlike denBroeder et al. (1959), did not assume that all weapons had the same probability of kill for any target  $j$ , but rather that each weapon's probability of kill for any target  $j$  was independent.

Julstrom (2009), Madni & Andrecut (2009), and Gelenbe et al. (2010) implement the MMR algorithm as a comparative benchmark in testing their heuristic approaches. Ahuja et al. (2007) utilize the MMR algorithm as one of three lower bounding schemes for their branch and bound algorithm.

### **Genetic Algorithms.**

There have been several implementations of the GA in the SWTA, each with a minor adjustment yet the same in structure and execution. Metler et al. (1990) was the first to implement the GA for the SWTA. Lee et al. (2002), Zhihua et al. (2009), Lee & Lee (2005), Bogdanowicz et al. (2013), Li et al. (2009), Fu et al. (2006), Lee et al. (2003), Lu et al. (2006), and Wu et al. (2008) are among the many subsequent researchers that utilized the GA for the SWTA.

### **VLSN.**

The very large scale neighborhood (VLSN) search metaheuristic is used by Ahuja et al. (2007) and Lee (2010) to improve upon informed feasible solutions. Their VLSN algorithms execute a heuristic search to efficiently find a quality solution and then they define local search neighborhoods within which to search for superior solutions.

### **Ant Colony Optimization.**

The Ant Colony Optimization (ACO) is another heuristic that is frequently implemented. It was first used by Lee et al. (2002), and Yanxia et al. (2008), Lee & Lee (2003), Shang (2003), Shang et al. (2007), Shang (2008), Huang & LI (2005), and Su et al. (2008) among others have used the ACO to solve the SWTA.

### **Other Heuristic Algorithms.**

Other techniques that do not fall into more generalized groupings have been demonstrated to efficiently find quality solutions to the SWTA. Day (1966) solves an integer relaxed NLP and utilizes rounding schemes. Wacholder (1989) implemented neural networks to find robust solutions. Ahuja et al. (2007) used a network flow based construction heuristic to find near optimal solutions to some of the larger



problems in the literature. Tokgöz & Bulkan (2013) compared the results of GA, Simulated Annealing (SA), Variable Neighborhood Search (VNS), and Tabu Search algorithms. Johansson & Falkman (2010) use Particle Swarm Optimization (PSO) and compare computational results to the GA, MMR, and exhaustive search algorithms. Similarly, Zeng et al. (2006) compares PSO with GA and a GA improved by greedy eugenics. Kwon et al. (1999) solves **S3** using a Lagrangian relaxation Branch and Bound Algorithm. Kline (2017) implemented the filtered beam search heuristic on **S1**, developed a heuristic based upon the optimal solution to the quiz problem and improved on these initial solutions using a metaheuristic which iteratively blocked assignment pairings which may have prevented superior solutions (Kline et al., 2017) and also developed a heuristic with similarities to the Hungarian Algorithm (Kline et al., 2017). See Hill & Pohl (2010) for a description of GA, SA, ACO, Tabu Search, and PSO.

### **DWTA.**

Less attention has been given to the DWTA as compared to the SWTA. Thus, there are fewer heuristic algorithms shared among researchers. Often, hybrid heuristic algorithms are used to inform one another in execution.

Metler et al. (1990) propose three greedy heuristics, the first of which is simply the MMR algorithm. In the second heuristic, the expected value of each pairing is computed and the selection of a random number determines the assignment based upon a probability mass function for which assignments with higher expected values have higher probabilities. The third heuristic proposed by Metler et al. (1990) is called the ALIAS Algorithm. This algorithm first updates the value of a target in stage  $t$  by dividing the value of the group by the total number of targets of type  $j$  in stage  $t$ . It then assigns weapons to targets based upon an MMR procedure,

updating the probabilities of kill and repeating until an assignment violates one of the constraints or the maximum number of iterations has occurred.

Chang et al. (1987) developed a heuristic algorithm for **D4** which utilizes a heuristic subroutine to solve the first stage. An iterative process then decrements the number of weapons to use in the first stage based upon its marginal contribution until the contribution is greater than some  $\epsilon$ , at which point the number of weapons for the first stage is fixed and the second stage is considered. This process iterates until either all weapons have been assigned or all stages have been considered. As a subroutine to solve the first stage, Chang et al. (1987) use three different heuristics: the MMR, an iterative linear network programming algorithm, and a nonlinear network flow algorithm.

Murphey (2000) develops a decomposition algorithm to solve **D9**. In this heuristic, he solves the first stage by some heuristic algorithm, saving the first stage solution and expected second stage solution. After this he solves the second stage primal and dual formulation across all possible second stage target outcomes. He uses these solutions to define the expected objective function value of the second stage. If this value exceeds the expected second stage objective function value previously determined, he adds a cut to the problem and repeats the process.

Xin et al. (2010) solve **D7** using Virtual Permutation (VP), TS, GA, and ACO. In a subsequent work, they developed a rule-based heuristic to solve **D7** in which they consider the saturation of the constraints in order to inform the greedy selection process by which they assign weapons to targets in a stage  $t$  (Xin et al., 2011).

Leboucher et al. (2013) use a Hungarian Algorithm to solve the assignment pairings for **D3a** and uses a GA-PSO hybrid algorithm to solve **D3b** in order to determine the firing order of the assignments. Khosla (2001) uses a GA-SA hybrid algorithm to solve **D8**. Chen et al. (2009) implements a GA to solve **D7**. Bertsekas et al. (2000)

uses Neuro-dynamic programming to obtain near optimal policies which he compares to optimal policies obtained through dynamic programming.

## 2.5 Discussion

### Evolution of WTA.

Research on the WTA has evolved since the work of Manne (1958) with developments in both the formulation of the problem and the solution techniques implemented. In the earliest works, reference to the limited capacity to solve large nonlinear problems (Day, 1966) resulted in attention on simplified formulations of the SWTA (denBroeder et al., 1959) and solution techniques capable given the computational capacity of the day (i.e., (Lemus & David, 1963), (Day, 1966)). Eckler & Burr (1972) proposed and discussed the possibility of solving dynamic variants of the SWTA but were unable to generate algorithms to solve such problems.

As computational power increased, so too did the ability to solve problems of increased complexity. Burr et al. (1985) modeled and solved one of the earliest DWTA problems, as did Chang et al. (1987), Soland (1987), and Hosein et al. (1988). Meanwhile, models of the SWTA with fewer assumptions were solved with novel approaches (i.e., (Wacholder, 1989), (Metler et al., 1990), (Kwon et al., 1999)).

This pattern continued into the 2000s, with model developments either capturing additional parameters which more closely resemble reality (i.e., (Shang et al., 2007) and (Karasakal, 2008)) or models which enabled faster optimal or near optimal solutions (i.e., (Malcolm, 2004), (Ahuja et al., 2007), and (Ahner & Parson, 2015)). Once developed, these models were solved using newer approaches (i.e., (Bertsekas et al., 2000), (Wu et al., 2008), (Kline et al., 2017)) or combinations of existing approaches which could be implemented efficiently (i.e., (Lee et al., 2002), (Ahuja et al., 2007), (Su et al., 2008), and (Xin et al., 2010)).

As computational power continues to grow, the WTA will likely continue to be the subject of research which improves upon existing solution techniques. Dynamic models which consider the time dependence of weapon utilization and target flight paths have been proposed ((Khosla, 2001) and (Leboucher et al., 2013)) but have received less attention than existing models. Improvements to the solution techniques in these models are yet to emerge, and as remarked by Khosla (2001), “in spite of the two-step approach [outlined in (Khosla, 2001)], each of the optimization problems still have a huge search space even for a modest number of threats, weapon systems, and time points.” Methods of improving on the two-step approach are yet to emerge. Similarly, Leboucher et al. (2013) remarks on the exponential growth of the problem and proposes a two-step solution technique, adding that an additional problem is “to be able to quantify the quality of one proposed solution.”

The future of the WTA will need to address the aforementioned difficulties of the scheduling-focused DWTa with techniques capable of exploiting the special structure of the problem. Additionally, there exist many parameters of the problem which are removed due to the increased computational complexity they would bring that could be introduced using novel modeling techniques.

### **Recent Developments.**

While the focus of the research discussed heretofore focuses primarily on the static and dynamic allocation of interceptors to offensive missiles, recent research has provided different frameworks through which this problem is addressed. We briefly discuss these recent developments here.

### **Sensor Weapon Target Assignment Problem.**

Missile defense depends on the accuracy and reliability of sensors to identify the type and position of each incoming missile so as to appropriately defend a protected asset. Much of the literature disregards the allocation of sensors and assumes the defender’s omniscience. However, different approaches concerning the consideration of a finite number of sensors are found within the literature.

Bogdanowicz & Coleman (2007) develop a model that seeks to maximize the sum of the benefits of assigning each sensor to each target and each weapon to each target. Zi-fen et al. (2011) combine the auction algorithm based technique developed by Bogdanowicz & Coleman (2007) to reduce the limitations that an imperfect network topology would introduce.

Others have considered the effect that sensors have on the probability of detecting incoming missiles. Jian & Chen (2015) models the damage probability of an interceptor as the probability that a sensor will identify the missile and the destructive capacity of the weapon with which the sensor is paired. Xin et al. (2018) extends this by modeling the probability of successful engagement as the product of the interceptor’s probability of kill and the sensor’s probability of detection.

### **Multi-objective Programs.**

Each of the formulations presented in §2 seeks to maximize the probability of destruction of all of the incoming missiles in some capacity. While the DWTA formulations include parameters and constraints that promote the preservation of some of the interceptors for subsequent salvos or subsequent shots to a leaker, solving each of these formulations results in the consumption of all available resources. Though it is important to defend a protected asset, there may be situations in which it is beneficial to conserve interceptors. As such, there has been research into the simultaneous

maximization of damage and minimization of shots.

Li et al. (2015) model the DWTa with an objective that simultaneously maximizes the expected damage and minimizes the ammunition consumption. They compare the performance of two solution techniques for this bi-objective program and later develop and compare a third technique (Li et al., 2017). Li et al. (2017) solve a similar formulation with a modified ant colony algorithm. Li et al. (2018) include a third objective which seeks to maximize the value of each weapon type and use a genetic algorithm to solve the multi-objective model.

### **Game Theory Approaches.**

While all of the research discussed thus far addresses the response to an adversary with no consideration of the adversary’s reaction, there has been research on this game theory aspect to missile defense. In contrast to the discussion regarding the research utilizing sensors and the research of multi-objective programs, the research of game theory approaches does not conform to similar models.

Shan & Zhuang (2013) develop a model that considers the impact of defensive resource allocation in the face of strategically focused and non-strategically focused adversaries. Golany et al. (2015) develop a model that seeks to place defensive resources in order to defend multiple assets and extend this model with a superior solution technique in Golany et al. (2017). Similarly, Boardman et al. (2017) models such a scenario and considers interceptor probabilities of kill.

Shalumov & Shima (2017) models a scenario wherein the protected assets are maneuvering aircrafts. Their model considers the flight paths of the aircrafts, the trajectories of the missiles, and the probabilities of kill of the interceptors in order to best guide the aircrafts and their defensive actions. Within the simulation they run, Shalumov & Shima (2017) test different assignment algorithms within a small scale

two agent game.

### **Alternate Applications.**

The WTA literature informs research beyond missile defense. Often, WTA works are cited for their modeling or solution techniques, as they are applicable in many assignment problems with quantifiable rewards or costs and limited resources. Gülpınar et al. (2018) framed their model and solution technique for a dynamic resource allocation problem on much of the same literature that is outlined in §2, §3, and §4 of this survey. Çetin & Esen (2006) model and solve a media allocation problem with an objective function which, if  $V_j$  is the audience type value,  $p_{ij}$  is the probability that audience  $j$  views advertisement  $i$ , and decision variable  $x_{ij}$  is the number of advertisements of type  $i$  to assign to audience  $j$ , is the formulation **S1**. Onay (2016) model neuromarketing with **S1** as an objective function where  $V_j$  is the value of the brain stimulus,  $p_{ij}$  is the probability that stimulus  $i$  affects the brain region  $j$ , and decision variable  $x_{ij}$  is the number of stimulants of type  $i$  to assign to brain region  $j$ . Another application using objective functions similar to **S1** is cancer treatment. The targeting of cancer cells with medication is modeled and solved by Çetin (2007) and Esen et al. (2008) using WTA research. Both Alighanbary (2004) and Bertuccelli & How (2011) model and solve unmanned aerial vehicle (UAV) assignment planning problems with static and dynamic WTA models. Lastly, Gelenbe et al. (2010) use WTA research to model and solve a problem of dispatching ambulances to emergencies with an objective function that is similar to formulations **S1** and **S3**.

### **Analysis of Literature Influence.**

Matlin (1970) put forth the first survey of the WTA literature, characterizing the problem with five components and defining elements of the problem which structured

its subsequent research. Due to the high volume of literature at present, a strategy to focus the considered literature for this survey was necessary, else an exhaustive list of the literature would obfuscate the state of the WTA and how it came to be. We considered the relevance of any work in the literature to be a function of its usefulness to subsequent research and used a rate of citation metric as a tool to limit our discussion heretofore.

**Table 1. WTA Literature by Citation Density**

Author	Year	Citations	Citation Rate	SWTA or DWTa	Exact or Heuristic
Lee, Z	2003	257	17.13	SWTA	Heuristic
Ahuja	2007	187	17	SWTA	Both
Lee, Z	2002	249	15.56	SWTA	Heuristic
Lee, Z	2005	117	9	SWTA	Heuristic
Lloyd	1986	230	7.19	SWTA	Exact
Xin	2011	50	7.14	DWTa	Heuristic
Xin	2010	55	6.88	DWTa	Heuristic
Karasakal	2008	57	5.7	SWTA	Exact
Ahner	2015	15	5	DWTa	Exact
Gelenbe	2010	38	4.75	SWTA	Heuristic
Chen	2009	42	4.67	DWTa	Heuristic
Bertsekas	2000	83	4.61	DWTa	Both
Rosenberger	2005	59	4.54	SWTA	Exact
Lee, M	2010	36	4.5	SWTA	Heuristic
Yanxia	2008	42	4.2	SWTA	Heuristic
Zeng	2006	50	4.17	SWTA	Heuristic
Wacholder	1989	119	4.10	SWTA	Heuristic
Lee, Z	2002	62	3.88	SWTA	Heuristic
Bogdanowicz	2013	16	3.2	SWTA	Heuristic
Eckler	1972	144	3.13	Both	Exact
Khosla	2001	52	3.06	DWTa	Heuristic
Murphey	2000	54	3	DWTa	Heuristic
Lee, Z	2003	43	2.87	SWTA	Heuristic
Johansson	2011	18	2.57	SWTA	Heuristic
Matlin	1970	115	2.40	SWTA	Exact
Madni	2009	21	2.33	SWTA	Heuristic
Hosein	1988	63	2.1	DWTa	Exact

As of September 2018

Table 1 shows that much of the work used in this survey with higher citation rate, given the current year of 2018, focuses on heuristic solutions to the SWTA. The entries within this table rate approximately 2 citations per year or more and demonstrate, by their consistent impact on research, the importance and substantial contribution they have made to the WTA literature.



At the same time, we find that this metric, while helpful in reducing the volume of literature to consider, can lead us to consider some works as less relevant due to the lower citation rate which is sensitive to original publication date. Despite this reduced rate, the works in Table 2 have a large number of citations and are foundational in much of the literature we consider highly relevant. As such, we include these works.

**Table 2. Included Foundational WTA Literature by Citation Count**

Author	Year	Citations	Citation Rate	SWTA or DWTA	Exact or Heuristic
Manne	1958	118	1.97	SWTA	Exact
Hosein	1989	46	1.59	DWTA	Exact
denBroeder	1959	83	1.41	SWTA	Exact
Soland	1987	31	1	DWTA	Exact
Day	1966	53	1.02	SWTA	Heuristic

As of September 2018

## 2.6 Conclusion

The WTA has a rich breadth of literature which serves to improve upon the theory and techniques necessary to efficiently solve these complex problems. Early works sought to find methods to transform the problem into a simpler form, assume many of the complexities away, or do both in order to generate a formulation which was manageable with the computational capacity of the day. The theories and techniques proposed by the earliest researchers, such as Manne (1958) and denBroeder et al. (1959), inform much of the current research and built a foundation upon which subsequent researchers were able to extend the theory and solution techniques of the WTA.

In this survey, we have provided nine static models and ten dynamic models for the WTA which have had an impact on the literature and have provided insights into the problem from a modeling perspective. Additionally, we have reviewed some of the exact algorithms, heuristic algorithms, and metaheuristic algorithms for the static and dynamic WTA. Some of these algorithms are widely used in the literature,

such as the branch and bound algorithm or the genetic algorithm. Others, such as the algorithm developed by Bogdanowicz (2012) or the rule based heuristic developed by Xin et al. (2010) were created to solve the WTA, efficiently exploiting the special structure of the problem.

The only consistent aspect of the WTA since its introduction into the field is its enduring relevance. As defensive strategies improve to enhance the capacity to mitigate the risk that ballistic missiles present, the technology of these ballistic missiles also improves. Additionally, while not addressed here, many non-defensive applications, in the business field and others, will continue to benefit from the lively research surrounding the weapon target assignment problem.

### III. A Heuristic and Metaheuristic Approach to the SWTA

#### 3.1 Introduction

The Weapon Target Assignment (WTA) Problem has been studied extensively in the field of operations research since its introduction to the field by Flood in 1957. It is the subject of many solution techniques that include exact algorithms, heuristic algorithms, and nature inspired metaheuristics. In much of the literature, a piecewise approximation of linear functions is used to transform the problem into one whose solution is more computationally accessible. While there are variants of the problem such as the dynamic WTA and two stage WTA, this paper focuses on the NP-complete Static WTA (SWTA) Problem.

Given  $n$  incoming targets, solving the problem results in the assignment of  $m$  weapons to the targets so as to minimize the collective residual expected value of the targets. The value of the targets,  $V_j$ , corresponds to their negative effects on the system being defended for  $j = 1, \dots, n$  targets and the number of weapons of type  $i$ ,  $w_i$ , which have an associated probability  $p_{ij}$  of destroying target  $j$ . As the problem seeks to minimize the residual value of targets, known in literature as target leakage, the probability of survival for an individual target is defined by  $q_{ij} = 1 - p_{ij}$ . The SWTA problem formulation is nonlinear and is defined by

$$\begin{aligned}
 & \min \sum_{j=1}^n V_j \prod_{i=1}^m q_{ij}^{x_{ij}} \\
 & \text{st} \sum_{j=1}^n x_{ij} \leq w_i \text{ for } i = 1, \dots, m, \\
 & x_{ij} \in \mathbb{Z}_+, \text{ for } i = 1, \dots, m, \ j = 1, \dots, n,
 \end{aligned} \tag{1}$$

where  $x_{ij}$  is the number of weapons of type  $i$  to assign to target  $j$ . In this paper,

we limit  $w_i = 1$  for  $i = 1, \dots, m$  for direct comparison with previous results given in Ahuja et al. (2007).

Solution techniques have expanded with regard to both the size of problem instances considered and the removal of simplifying assumptions. Manne (1958), apart from contributing the first model of this problem to the literature, identified an optimal solution to the problem when two “not-so heroic approximations” are introduced: all weapons have the same probability of kill for any one target, and the decision variables,  $x_{ij}$ , are binary-valued. Manne whereas demonstrated how a small toy problem can be optimized by observation, denBroeder et al. (1959) offers the first algorithm capable of solving larger instances using Manne’s assumptions. Known in the literature as the Maximum Marginal Return (MMR) algorithm, denBroeder sequentially assigns weapons to targets that respectively provide a maximum marginal contribution to the objective function. Kolitz (1988) further defined the conditions by which the MMR algorithm is optimal. Removing the assumption that all weapons have the same probability of kill for a particular target, Chang et al. (1987) limited the number of weapons assigned to each target to be at most one and, by utilizing a linear approximation of the objective function, was able to solve the problem via an iterative linear network programming algorithm (ILINE) that assigns the best weapon-target pairings and updates the number of available weapons at each iteration. The limitation on the number of weapons assigned to each target was removed by Kwon et al. (1999) and Ahuja et al. (2007) to optimize the problem using a branch and bound algorithm. These branch and bound approaches respectively utilized Lagrangian relaxation techniques (Kwon) and linear transformations (Ahuja) in their execution. Johansson & Falkman (2009) sought to define the limits of instance sizes for which a brute force algorithm can optimize the nonlinear problem, and Kline (2017) developed a branch and bound algorithm to optimize larger instances of the nonlinear problem.

As the WTA problem has been proven to be NP-complete (Lloyd & Witsenhausen, 1986), many heuristic algorithms have been utilized to find suboptimal solutions. Neural networks (Wacholder, 1989), genetic algorithms (e.g., Lee et al., 2003, 2002; Shang et al., 2007), ant colony algorithms (Lee et al., 2002), and particle swarm algorithms (Zeng et al., 2006) are but a few of the heuristic and metaheuristic solution techniques applied.

Kline (2017) proved the convexity of the untransformed SWTA objective function, developed methods of finding exact solutions to smaller problem sizes, and both created and tested a new heuristic capable of finding quality solutions in real time. This heuristic, while finding solutions within hundredths of a second for large instances (e.g.,  $n = 160$ ,  $m = 80$ ) of these NP-hard problems, usually does not guarantee optimal solutions. For smaller instances (e.g.,  $n \leq 10$ ,  $m \leq 10$ ), the heuristic attained solutions that were approximately 5% from the global optimum, on average, motivating this study to improve the solution procedure.

In this paper, we review the Modified Quiz Problem (MQP) heuristic from Kline (2017) in §2. In §3, we introduce a new metaheuristic which invokes the MQP heuristic as a subroutine which we iterate while preventing judiciously selected assignments to search for improved assignment pairings. We present a comparison of the results of this metaheuristic with the well regarded construction heuristic developed by Ahuja et al. (2007) in §4 and provide conclusions in §5.

### 3.2 Quiz Problem Heuristic

The heuristic that serves as a kernel within the metaheuristic examined herein is based upon the parametric characteristics of the optimal solution to the Quiz Problem. Bertsekas & Castañon (1999) first applied the solution characteristics to the Quiz Problem within a heuristic search for scheduling problems, and Ahner (2005)

used the approach to route unmanned aerial vehicles in risky environments.

The Quiz Problem states that an individual presented with a series of questions  $u = 1, \dots, n$  with value  $v_u$  has probability  $p_u$  of correctly answering a question. Further, the individual answers questions, receiving the value of each question as a reward, until he answers one incorrectly, at which point the quiz is terminated. The objective of the Quiz Problem is to identify the order in which to answer the questions to maximize the value of the collective reward. Bertsekas & Castañón (1999) showed that the strategy to maximize return is an index policy wherein questions are answered in descending values of  $y_u$ , which is defined as

$$y_u = \frac{v_u p_u}{1 - p_u} = \frac{v_u p_u}{q_u}.$$

For the SWTA, we use Ahner’s residual value strategy to define the value of each weapon-target assignment as  $y_{ij}^0$ , which allows us to select the maximum return for a weapon-target assignment  $x_{ij}$ , where

$$(\hat{i}, \hat{j}) \in \arg \max_{\substack{i=1, \dots, m, \\ j=1, \dots, n}} \{y_{ij}^0\}.$$

We then redefine our target value to be  $V_j \leftarrow V_j q_{i\hat{j}}$ , which is the residual value of the selected target given the weapon assigned. We also redefine our probabilities of kill for the selected weapon  $w_i$  to be  $p_i = 0$ , as we have only one weapon of each type for our models (W.L.O.G., an instance having more than one weapon of a given type can be modeled with our construct by considering the multiple weapons of the same type as multiple weapons types having identical capabilities and only one weapon each). Using these updated values, we update our value array to be  $y_{ij}^1$ . We repeat this process until each of the weapons is assigned to a target.

Figure 1 shows the Modified Quiz Problem Heuristic. The heuristic initializes

the assignment solution,  $\mathbf{x}$ , to an  $n \times m$  zeros matrix for a problem instance having  $n$  weapons and  $m$  targets. As referenced above, we build a value array  $\mathbf{y}$ , which is defined by

$$y_{i,j}^0 = \frac{V_j p_{ij}}{q_{ij}}.$$

We then identify the two largest values in  $\mathbf{y}$ , which we denote as  $y(i_1, j_1)$  and  $y(i_2, j_2)$ . If  $i_1 \neq i_2$ , we check to see whether the greedy assignment is preferred according to a process defined in the following paragraph. If  $i_1 = i_2$ , we set  $x(i_1, j_1) = 1$  and increment a counter  $k$  by 1. If our counter is equal to the number of weapons after this increment, we terminate the heuristic. Otherwise, we update our value array  $\mathbf{y}$  by redefining  $V_{j_1} \leftarrow V_{j_1} q_{i_1 j_1}$  and setting the probabilities of weapon  $i_1$  to zero, (i.e.,  $p_{i_1 j} = 0, j = 1, \dots, n$ ).

We note a shortcoming for greedy selection-based heuristics. These heuristics seek to define pairings based upon the greatest expected value of the weapon-target assignment, that is  $V_j p_{ij}$ , which aligns with the conceptual underpinnings of a greedy algorithm. The shortcoming in this approach can be illustrated by examining a case having two weapons and two targets, with target values  $\{V_1, V_2\}$  and probabilities of kill  $\{p_{11}, p_{12}, p_{21}, p_{22}\}$ . If we define the greatest expected value to be

$$\max \left( \begin{bmatrix} V_1 p_{11} & V_2 p_{12} \\ V_1 p_{21} & V_2 p_{22} \end{bmatrix} \right) = V_1 p_{11},$$

and we further assume that, for this case,

$$V_1 p_{11} - V_2 p_{12} < V_1 p_{21} - V_2 p_{22},$$

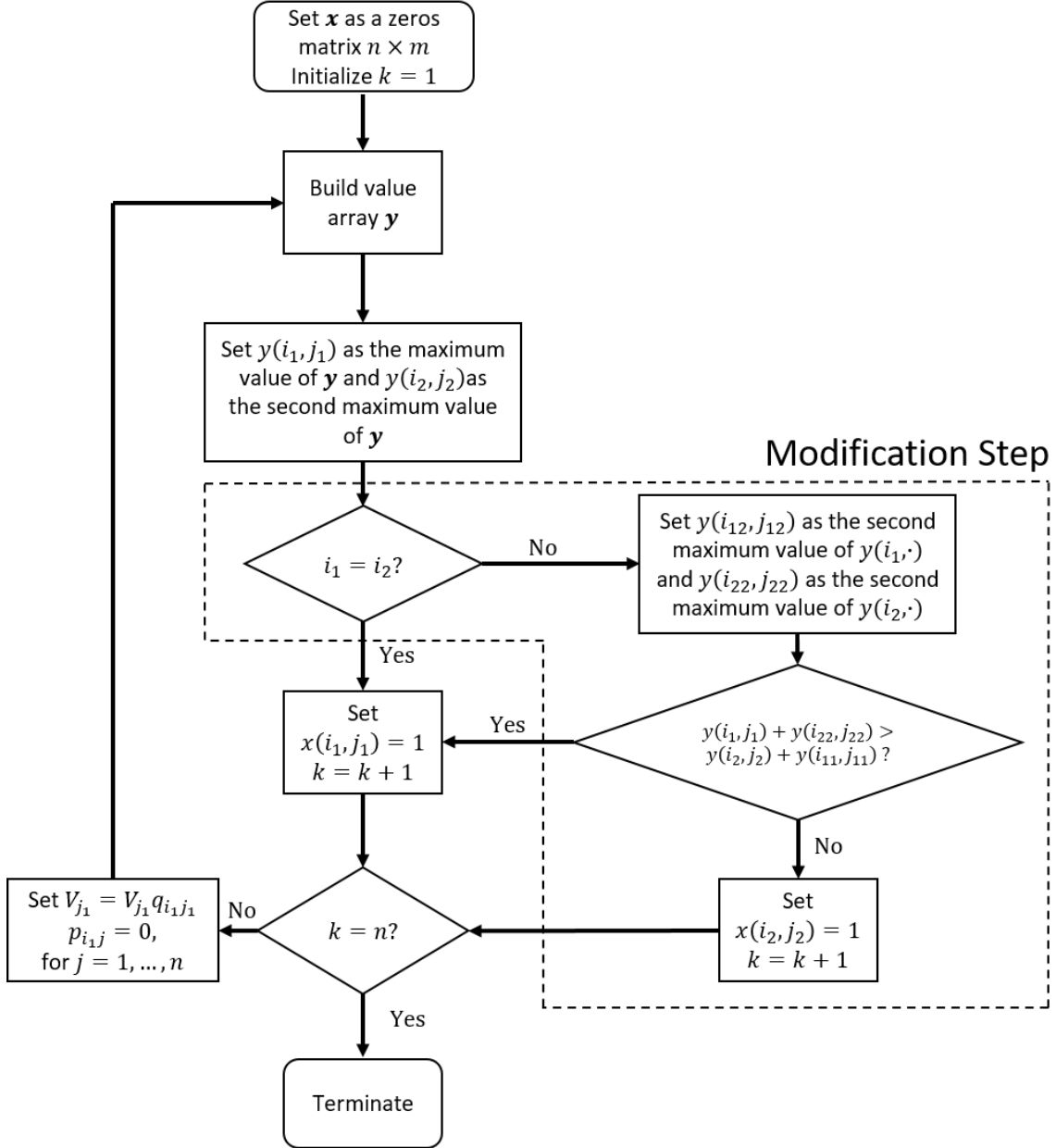


Figure 1. Quiz Problem Heuristic and Modified Quiz Problem Heuristic



we can see that, with a simple rearrangement of the above inequality, we have

$$V_1p_{11} + V_2p_{22} < V_1p_{21} + V_2p_{12}.$$

Thus, selecting the pairing with the greatest expected value will result in a lower final solution value than selecting the alternative. We incorporate into the Quiz Problem Heuristic a step that checks whether

$$y_{ab} - \max_{j \neq b} \{y_{aj}\} < y_{cd} - \max_{j \neq d} \{y_{cj}\},$$

where  $y_{ab} > y_{cd} > \dots > \min_{i,j} \{y_{ij}\}$ . In the case wherein the above inequality holds true, we select  $y_{cd}$  rather than  $y_{ab}$  as our assignment within such an iteration.

Although we may find superior solutions with this modification, we note that this additional routine increases the computational requirement of the heuristic. We will show in §3 that the Eminent Domain Metaheuristic proposed herein similarly resolves the shortcoming of implementing a greedy selection process through repetition of a subroutine, so we will not invoke the modification step in the metaheuristic proposed in §3 or in the Quiz Problem Heuristic in §4.

### 3.3 Eminent Domain Metaheuristic

Although the Quiz Problem Heuristic has an appealing simplicity for implementation and requires relatively little computational effort, even when utilized to solve large problem instances, the solution quality can be sufficiently degraded relative to an accepted benchmark heuristic (e.g., see Kline (2017)) to merit the consideration of an alternative approach. As such, a metaheuristic capable of using the solution from the QP heuristic as an initial feasible solution has the potential to exceed current top performing heuristics in both computational time and solution quality. Herein,

we develop the Eminent Domain (ED) Metaheuristic that, given an initial feasible solution, iteratively denies the assignment pairings of a subset of weapon-target pairs that yield the most value for the benefit of other potential assignments.

Upon examining the QP heuristic, we found that inferior solutions are identified when an assignment pairing is made that causes the prevention of ultimately superior pairings, effectively converging towards a local-but-not-global optimal solution. That is, for an instance in which  $i - l$  and  $k - j$  is part of a global optimal solution, such a local optimum convergence may occur if the assignment of weapon  $i$  to target  $j$  has a superior expected value compared to assigning weapon  $i$  to target  $l$ , but where a  $k \rightarrow l$  assignment is far worse than a  $k - i$  assignment. We initially considered using a neighborhood search metaheuristic to improve upon the initial solution, but since there is no guarantee that assignments must be spread evenly (i.e., there can be unassigned targets even when the number of weapons equals the number of targets), this technique would have adopted the logical construct of a full enumeration approach, negating rather than exploiting the computational efficiency of the QP heuristic. We found that, as an alternative, running the QP heuristic while instilling a side constraint iteratively preventing the best assignments would allow for the heuristic to obtain an improved objective value.

The formalized process by which we execute the ED Metaheuristic, as depicted in Figure 2, is as follows. We first find a feasible solution using the QP heuristic. We next define a method by which we assign values to the assignments of our initial solution. From this solution, we define a threshold value above which we iteratively implement our QP heuristic with side constraints while preventing such assignments. The best solution found is returned using a termination rule where either the metaheuristic is run to completion or a maximum computation time is exceeded.

We seek a method for alternatively defining the quality of each assignment because

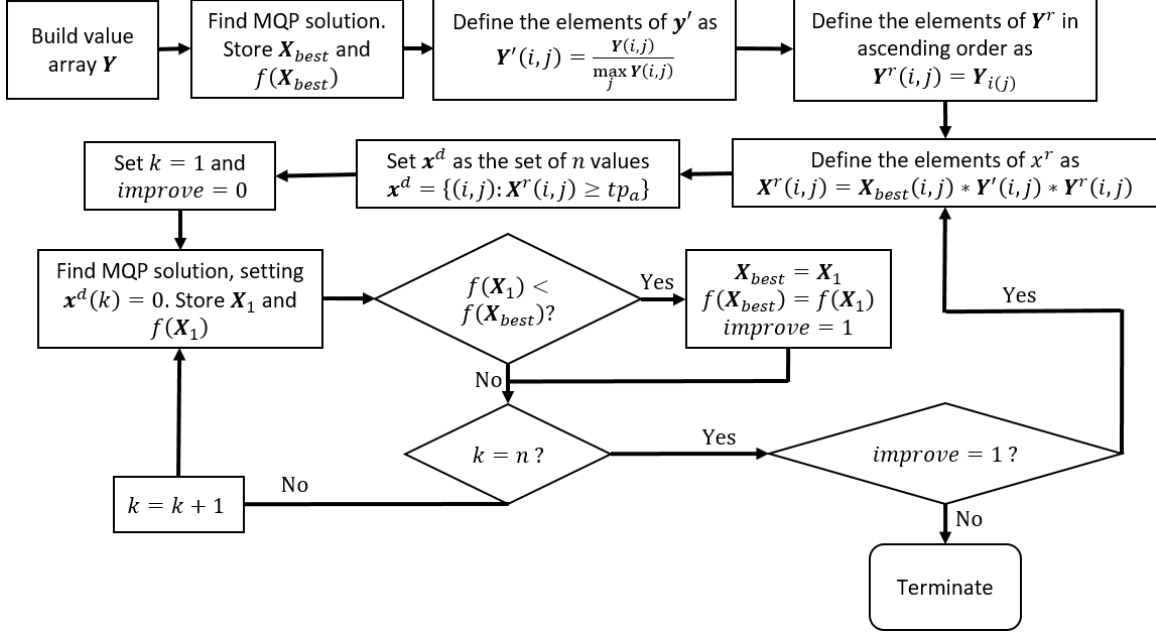


Figure 2. Eminent Domain Metaheuristic

the *decay* of assignment utility in one weapon may far exceed that of another. Take, for example, the values of the following problem with two weapons and two targets:

$$\begin{bmatrix} V_1 p_{11} & V_2 p_{12} \\ V_1 p_{21} & V_2 p_{22} \end{bmatrix}$$

where  $V_1 p_{11} = 10$ ,  $V_1 p_{21} = 9$ ,  $V_2 p_{12} = 8$ , and  $V_2 p_{22} = 6$ . In assigning Weapon 1 to Target 1, which follows from a greedy selection process, we force the assignment of Weapon 2 to Target 2. The resulting sum of assignment values is 16, which is inferior to that of assigning Weapon 1 to Target 2 and Weapon 2 to Target 1, which has a sum of assignment values of 17. We note that the decay of assignment values of Weapon 1 is less than that of Weapon 2. If we consider this assignment value decay, we can observe the quality of the assignment of Weapon 2 to Target 2 to be lower than that of Weapon 1 to Target 2. While this is similar to the greedy selection shortcoming illustrated in §2, it is worth redefining because the assignment value

decay is the basis for the ED Metaheuristic. In the following paragraphs, we describe two alternatives for defining the quality of each assignment to block myopically higher-valued assignments and allow the subroutine to make assignments that may be part of a solution having a lower objective function value.

The first method to define the quality of assignments considers the relative value decay of assignments for each weapon. As such, we denote this the Ratio-based Method. We first define the values of each possible assignment in keeping with the MQP heuristic:  $\mathbf{y}_{ij} = \frac{V_j p_{ij}}{q_{ij}}$ . We then compute the relative values for each weapon, or

$$\mathbf{y}'_{ij} = \frac{\mathbf{y}_{ij}}{\max_j \{\mathbf{y}_{ij}\}},$$

where  $\mathbf{y}'_{ij} \in (0, 1]$ . Further, we define the rank order of each weapon's assignments,  $\mathbf{y}^r$ , with 1 being the worst and  $t$  being the best. We compute an element-wise product of the rank order matrix with the relative value matrix to find a ratio value matrix. Next, we multiply this element-wise by our initial solution,  $\mathbf{x}_0$ , resulting in an  $m \times n$  matrix of assignment scores,  $\mathbf{x}^r$ . Table 3 illustrates each step of the Ratio-based ED Metaheuristic process through an example.

Next, we define a threshold percentage,  $p_a$ , which identifies the number of assignments to iteratively deny. We define the set of assignments which we will deny,  $\mathbf{x}^d$ , to be the set of all  $\mathbf{x}^r \geq t p_a$ . For example, we define  $p_a = 0.8$ , we would find each assignment score  $\mathbf{x}^r \geq 0.8t = 4$  and run the subroutine 2 times, iteratively blocking assignments

$$w_3 - t_2$$

$$w_5 - t_3$$

We allow for an increased number of denials so that we may consider the solution

**Table 3. Ratio-Based Assignment Scores Computation**

$\mathbf{y}$	$j = 1$	2	3	4	5
$i = 1$	151.93	228.05	139.92	79.33	101.02
2	154.53	119.94	176.97	105.88	53.91
3	99.02	259.72	143.02	61.12	58.39
4	103.89	231.80	118.25	147.48	179.99
5	120.18	335.10	442.78	83.84	59.07
$\mathbf{y}'$	$j = 1$	2	3	4	5
$i = 1$	0.67	1	0.61	0.35	0.44
2	0.87	0.68	1	0.60	0.30
3	0.38	1	0.55	0.24	0.22
4	0.45	1	0.51	0.64	0.78
5	0.27	0.76	1	0.19	0.13
$\mathbf{y}^r$	$j = 1$	2	3	4	5
$i = 1$	4	5	3	1	2
2	4	3	5	2	1
3	3	5	4	2	1
4	1	5	2	3	4
5	3	4	5	2	1
$\mathbf{x}_0$	$j = 1$	2	3	4	5
$i = 1$	1	0	0	0	0
2	0	0	0	1	0
3	0	1	0	0	0
4	0	0	0	0	1
5	0	0	1	0	0
$\mathbf{x}^r$	$j = 1$	2	3	4	5
$i = 1$	2.66	0	0	0	0
2	0	0	0	1.20	0
3	0	5.00	0	0	0
4	0	0	0	0	3.11
5	0	0	5.00	0	0

where single denials, single-and-pairwise denials, etc. are implemented. For example, if we were to allow for up to two simultaneous denials, the blocked assignments would include

$$\begin{aligned} w_3 - t_2 \\ w_5 - t_3 \\ w_3 - t_2 \ \& \ w_5 - t_3 \end{aligned}$$

We store the objective function value and assignment matrix of the best solution and, if we find an improved solution, we run this process again. This metaheuristic terminates when no improvements are found during an iteration of the denial process.

A second method we develop to define the quality of each assignment is similar to the first though, instead of relying on relative values, we use the least squares method to find the slope for the solution value decay of each weapon. We call this method the Slope-based Method. Referring to Table 3, this alternative method replaces the second process where we define the  $\mathbf{y}'$  matrix. We divide each vector of slopes by the minimum slope to find a scaled vector of slopes  $m_i \in (0, 1]$ , which we subsequently multiply by the rank of each assignment, as in the Ratio-based Method with the relative value of each weapon. Table 4 illustrates the steps of the Slope-based ED Metaheuristic through the same example as illustrated in Table 3.

### **Improvements on the ED Metaheuristic.**

We define a metaheuristic capable of finding improved solutions using a heuristic subroutine. The implementation of this metaheuristic only improves the heuristic search algorithm solution. Further, the implementation of the ED Metaheuristic with single-and-pairwise denials will not find inferior solutions to those found using

Table 4. Slope-Based Assignment Scores Computation

$\mathbf{y}$	$j = 1$	2	3	4	5
$i = 1$	151.93	228.05	139.92	79.33	101.02
2	154.53	119.94	176.97	105.88	53.91
3	99.02	259.72	143.02	61.12	58.39
4	103.89	231.80	118.25	147.48	179.99
5	120.18	335.10	442.78	83.84	59.07
$\mathbf{y}^r$	$j = 1$	2	3	4	5
$i = 1$	4	5	3	1	2
2	4	3	5	2	1
3	3	5	4	2	1
4	1	5	2	3	4
5	3	4	5	2	1
	m	Slope	Relative Slope		
$i = 1$		-34.83	0.34		
2		-29.48	0.29		
3		-48.46	0.48		
4		-31.76	0.31		
5		-101.87	1		
$\mathbf{y}'$	$j = 1$	2	3	4	5
$i = 1$	1.37	1.71	1.03	0.34	0.68
2	1.16	0.87	1.45	0.58	0.29
3	1.43	2.38	1.90	0.95	0.48
4	0.31	1.56	0.62	0.94	1.25
5	3.00	4.00	5.00	2.00	1.00
$\mathbf{x}_0$	$j = 1$	2	3	4	5
$i = 1$	1	0	0	0	0
2	0	0	0	1	0
3	0	1	0	0	0
4	0	0	0	0	1
5	0	0	1	0	0
$\mathbf{x}^r$	$j = 1$	2	3	4	5
$i = 1$	1.37	0	0	0	0
2	0	0	0	0.58	0
3	0	2.38	0	0	0
4	0	0	0	0	1.25
5	0	0	5.00	0	0

only a single denial.

**Theorem 1** *The single-and-pairwise denials variant of the ED Metaheuristic finds non-degrading solutions relative to the single-denial variant, which finds non-degrading solutions to its subroutine QP Heuristic.*

PROOF. Let  $\mathbf{X}^{QP}$  be the feasible solution found by the QP heuristic. Further, let  $\mathbf{X}^{(1)}$  be the set of solutions found by the ED Metaheuristic with one denial. As  $\mathbf{X}^{(1)}$  is a finite set, there exists a solution  $\mathbf{x}^* \in \mathbf{X}^{(1)}$  for which  $f(\mathbf{x}^*) \leq f(\mathbf{x})$ ,  $\forall \mathbf{x} \in \mathbf{X}^{(1)}$ . As illustrated in Figure 2, if  $f(\mathbf{x}^{QP}) < f(\mathbf{x})$ ,  $\forall \mathbf{x} \in \mathbf{X}^{(1)}$ ,  $\mathbf{x}^* = \mathbf{x}^{QP}$ . Otherwise,  $f(\mathbf{x}^*) < f(\mathbf{x}^{QP})$ .

Similarly, let  $\mathbf{X}^{(2)}$  be the set of all solutions found by the ED Metaheuristic with two simultaneous denials. Let  $\mathbf{x}_r \in \mathbf{X}^{(1)}$  be the solution found when blocking the  $r^{th}$  assignment and  $\mathbf{x}_{r,s} \in \mathbf{X}^{(2)}$  be the solution found when blocking both the  $r^{th}$  and  $s^{th}$  assignments. Then  $\forall \mathbf{x}_r, \exists \mathbf{x}_{r,s}$  such that  $\mathbf{x}_r = \mathbf{x}_{r,s}$  when  $r = s$ . Therefore,  $\mathbf{X}^{(1)} \subset \mathbf{X}^{(2)}$ . As above, there exists a solution  $\mathbf{x}^* \in \mathbf{X}^{(2)}$  for which  $f(\mathbf{x}^*) \leq f(\mathbf{x})$ ,  $\forall \mathbf{x} \in \mathbf{X}^{(2)}$ .

From these, we can assert that  $\forall \mathbf{x}^1 \in \mathbf{X}^{(1)}$  and  $f(\mathbf{x}^2) \leq f(\mathbf{x}^1) \leq f(\mathbf{x}^{QP})$ ,  $\forall \mathbf{x}^2 \in \mathbf{X}^{(2)}$ . □

### 3.4 Computational Results

We test each heuristic and metaheuristic to solve a set of instances by designing random parameters within various instance sizes ranging from 5 weapons and 5 targets to 80 weapons and 160 targets. We consider 15 problem sizes, shown in Table 18, which affix the number of weapons and targets for a set of randomly generated instances for each size. For each target value, a uniformly distributed continuous variable value [25, 100] is randomly generated, and we also assigned randomly generated probabilities of kill as uniformly distributed continuous variables [0.6, 0.9] so



that each weapon has a different probability of kill for each target. This allows us to compare results to Ahuja et al. (2007), who used the same distributions. We generate 20 problem instances for each of the 15 problem sizes and perform all tests on a computer having an Intel Xeon E5-2650 v2 processor with 128 GB RAM. Each solution method is applied to the same set of 20 problem instances for each problem size to enable a comparison of solution methods’ relative effectiveness and computational efficiency.

**Table 5. Test Problem Sizes**

Problem Size	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Weapons	5	10	10	10	15	20	20	20	40	40	40	40	80	80	80
Targets	5	5	10	20	10	10	20	40	10	20	40	80	40	80	160

### **Comparison of Heuristics.**

The baseline heuristic, the QP heuristic, is compared to the Construction heuristic developed by Ahuja et al. (2007). The motivation for introducing a modification to the Quiz Problem (QP) heuristic is explained in §2 and generally results in improved objective values. However, this modification increases the computational requirement of the heuristic. As the ED Metaheuristic similarly resolves the shortcoming of implementing a greedy selection process through repetition of the subroutine, we demonstrate results using the QP heuristic without the modification.

Table 6 reports the performance of the QP and Construction Heuristics regarding both the number of instances for which each heuristic dominates in terms of objective function value, and the average required computational effort for each problem size. We observe that, whereas the QP heuristic attained the superlative objective function value for 79 of the 300 (26%) test instances, its required computational effort was notably better than the Construction Heuristic. Despite attaining the superlative

objective function value for 77% of the instances tested (including some instances for which both heuristics attained an identical solution), the Construction Heuristics required computational effort is problematic for practical use. By comparison, the QP Heuristic required 99.96% less computational effort, on average, across problem sizes. Moreover, for Problem Sizes 9-15, the Construction Heuristic required over 15 seconds and up to 22 minutes, on average, for the implementation of the WTAP solution to defend against incoming missiles. Such solution times will not allow enough time to act on the decision in a timely manner.

### **Improvements with ED Metaheuristic.**

In this section, we test and present the improved solutions found via the ED Metaheuristic and conduct selected sensitivity analyses for key algorithmic parameters.

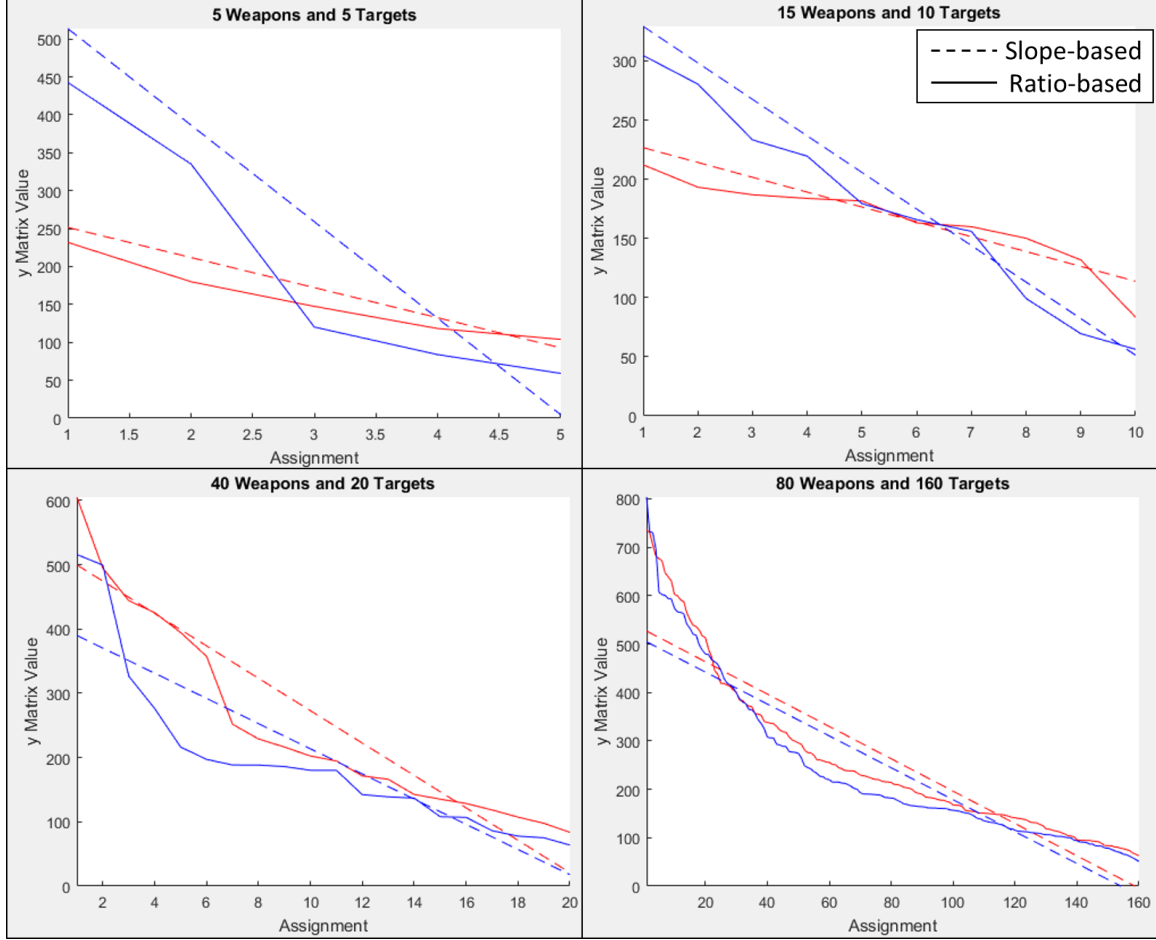
During initial testing for which we do not explicitly display the results herein, we observed that the improvements for smaller problem sizes are greater with the Ratio-based ED, whereas they are greater with the Slope-based ED for larger problem sizes. As problem size increases, more targets are available for each weapon. This results in an increased density of assignment values about the mean. Because the Slope-based ED computes a least squares model of a weapons assignment quality decay, the increased density about the mean reduces the slope and decreases the intercept relative to the models of smaller problems. As the heuristic begins assigning weapons to targets, the steep immediate decay of assignment pairings for larger problems quickly reduces to a far slower decay whereas the slope of the model gives a constant decay. This means that, for larger problems, the assignment values given by Ratio-based ED will have very high scores for the top assignments and the rest will be relatively small whereas Slope-based ED will have steadily decreasing scores. This yields a greater quantity of assignments in our denial set  $\mathbf{x}^d$  for the Slope-based ED

**Table 6. Comparison of QP Heuristic and Construction Heuristic Over 20 Instances per Problem Size**

Problem			Best Solutions		Average Time (sec)	
Size	Weapons	Targets	QP	Construction	QP	Construction
1	5	5	5	19	0.000254	0.224
2	10	5	5	19	0.000382	0.445
3	10	10	5	17	0.000386	0.441
4	10	20	7	13	0.000406	0.487
5	15	10	4	16	0.000538	0.767
6	20	10	0	20	0.000768	1.608
7	20	20	1	19	0.000815	2.208
8	20	40	14	6	0.00121	4.455
9	40	10	1	19	0.00136	17.997
10	40	20	2	18	0.00210	20.569
11	40	40	1	19	0.00250	14.311
12	40	80	14	6	0.00352	47.077
13	80	40	0	20	0.00704	309.641
14	80	80	0	20	0.00876	177.106
15	80	160	20	0	0.0117	1348.815

vis-à-vis the Ratio-based ED for larger problem sizes.

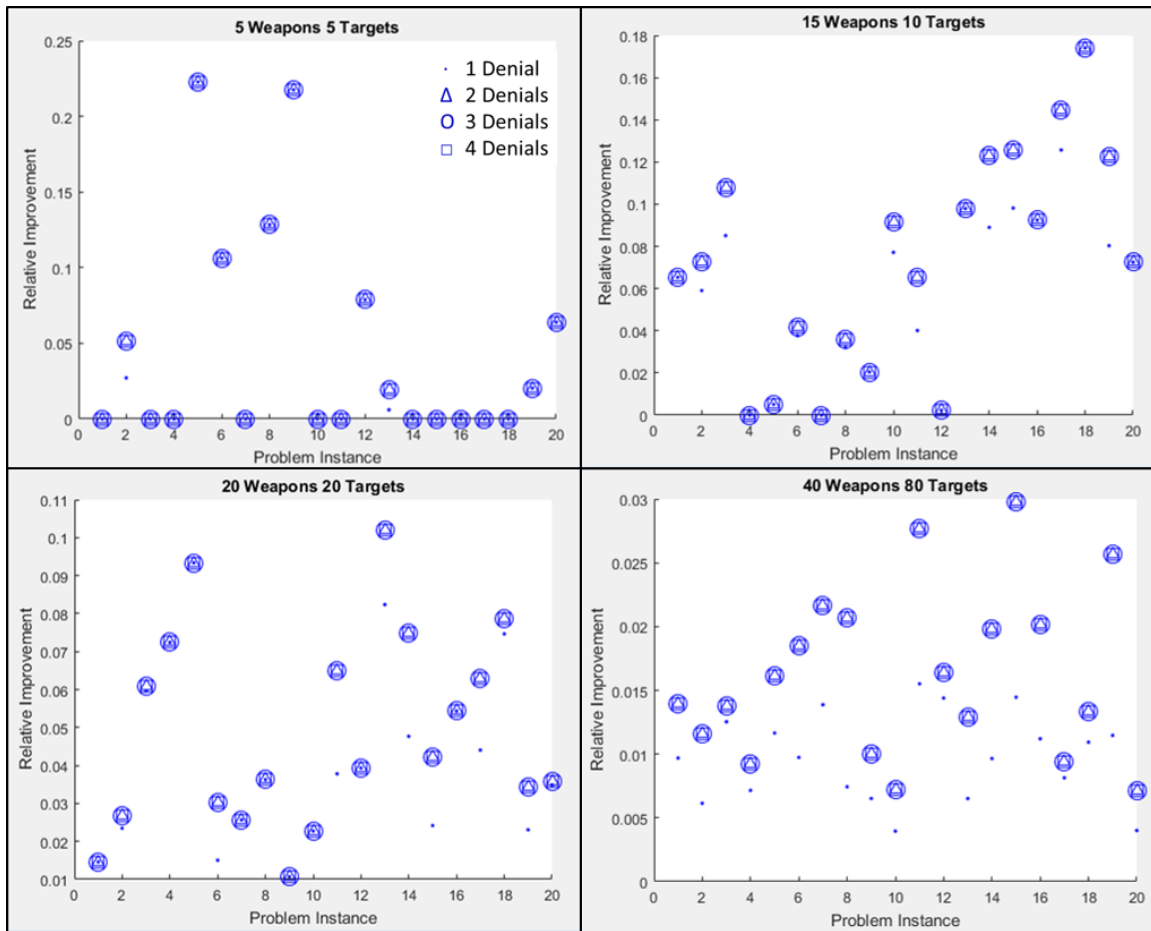
For smaller problem sizes, we observe that our Ratio-based ED improves the solution better than the Slope-based ED. This is due to the increased variation between weapon assignment decay slopes. With the larger problems, the assignment decay is very similar among each of the weapons, but with smaller problems this is not the case. As such, the Slope-based ED, which normalizes the slopes, is more likely to have one weapon with a relatively large slope which corresponds to smaller normalized slopes for the other weapons, and ultimately results in lower assignment quality scores for those weapons. As the Ratio-based ED does not consider the assignment decay of each weapon relative to each other weapon, its scores are not impacted by outliers. Figure 3 shows the assignment quality decay for two instances each for four problem sizes. The solid line depicts the actual decay of assignment values, whereas the dotted line shows the slope of the decay for the given weapons. This illustrates the variation between slopes and the density of assignment values about the mean.



**Figure 3. Slope-based ED and Ratio-based ED Assignment Value Decay Rates for 2 Instances Each of 4 Different Problem Sizes**

We conducted additional testing on the ED Metaheuristic using up to one, two, three, or four simultaneous denials, respectively, to identify the impact on solution quality. Figure 4 illustrates the relative improvements for each of the 20 instances for four different problem sizes, using the Slope-based ED with  $p_a = 0.4$ , wherein, e.g., an improvement of 0.11 indicates an 11% relative improvement in the reported objective function value. We observe that there is no benefit in these problems to allowing three and four simultaneous denials. In comparing our improved solutions to known optimal solutions for smaller problem sizes, we note that some problems are improved by denying only one assignment of one weapon and multiple assignments

of another weapon, a process the ED Metaheuristic does not perform due to the combinatorially-influenced growth in the computational effort it would entail.



**Figure 4. Relative Improvements Using Multiple Simultaneous Denials in the Slope-based ED Metaheuristic (with  $p_a = 0.4$ )**

As evidence to this conjecture, we note in Table 7 that the average computational effort required for these multiple denials grows exponentially for medium and larger-sized problem instances. We observe the points at which the ED Metaheuristic variants with “up to 3” and “up to 4” multiple denials each exceed the computational time required for the Construction Heuristic to find a solution.

We test the improvements made by the Slope-based ED Metaheuristic while varying the threshold percentage. We run each of the 20 instances using threshold percentages  $p_a = \{0, 0.25, 0.5, 0.75, 1\}$  and examine the average relative effect on the QP

**Table 7. Average Computational Effort for Multiple Denials in the ED Metaheuristic**

Weapons	Targets	Average Time (sec)				Construction
		ED With Simultaneous Denials				
		1	up to 2	up to 3	up to 4	
5	5	0.00955	0.00521	0.00671	0.00876	0.224
15	10	0.0199	0.0806	0.336	1.210	0.767
20	20	0.0373	0.253	1.528	7.650	2.208
40	80	0.328	6.806	92.499	992.206	47.077

Heuristic solution and the average computational time, respectively. Table 8 reports the improvements and computational times with one denial. The best improvements in average solution quality occur with a threshold percentage of 0%, which is expected as it ensures that we deny each such assignment rather than a subset of our assignments.

**Table 8. Improvements for 1 Denial With Different Threshold Percentages for the Slope-based ED Metaheuristic**

Weapons	Targets	Relative Improvement (%)					Average Time (sec)				
		Threshold Percentage					Threshold Percentage				
		0	0.25	0.5	0.75	1	0	0.25	0.5	0.75	1
5	5	3.935	2.683	2.683	2.683	2.683	0.00353	0.00334	0.00325	0.00317	0.00319
10	5	4.672	4.199	4.199	4.199	4.199	0.00729	0.00683	0.00602	0.00702	0.00596
10	10	5.728	5.160	5.160	4.525	4.525	0.00730	0.00637	0.00612	0.00814	0.00600
10	20	1.864	1.864	1.423	1.273	1.273	0.00762	0.00769	0.00656	0.00607	0.00609
15	10	5.415	4.647	3.929	3.929	3.929	0.0111	0.00966	0.00890	0.00891	0.00889
20	10	5.972	3.338	3.338	3.338	3.338	0.0151	0.0119	0.0118	0.0118	0.0117
20	20	3.187	2.323	2.323	2.323	2.323	0.0155	0.0120	0.0129	0.0120	0.0120
20	40	1.158	1.158	1.088	0.762	0.762	0.0186	0.0190	0.0193	0.0166	0.0131
40	10	3.805	2.892	2.892	2.892	2.892	0.0301	0.0236	0.0235	0.0237	0.0234
40	20	2.396	1.708	1.708	1.708	1.708	0.0346	0.0277	0.0286	0.0280	0.0263
40	40	2.192	2.054	1.853	1.853	1.853	0.0378	0.0291	0.0284	0.0272	0.0276
40	80	0.803	0.803	0.803	0.638	0.638	0.0488	0.0504	0.0513	0.0324	0.0325
80	40	2.659	2.152	1.999	1.999	1.999	0.0929	0.0680	0.0700	0.0625	0.0600
80	80	1.413	0.927	0.841	0.841	0.841	0.135	0.0845	0.0759	0.0760	0.0741
80	160	0.386	0.386	0.386	0.349	0.349	0.196	0.195	0.195	0.115	0.110

Within Table 9, we note that varying the threshold percentage is far more computationally expensive when we allow for two simultaneous denials, but that the improvements to the QP solution are far greater. We can observe that while all of

the average solution improvements increase (save those of a threshold percentage of 100%) when allowing two simultaneous denials, the improvements between  $p_a = 0.25$  and  $p_a = 0.5$  increase the most, on average.

**Table 9. Improvements for 2 Denials With Different Threshold Percentages for the Slope-based ED Metaheuristic**

Weapons	Targets	Relative Improvement (%)					Average Time (sec)				
		Threshold Percentage					Threshold Percentage				
		0	0.25	0.5	0.75	1	0	0.25	0.5	0.75	1
5	5	5.845	5.845	5.447	5.238	2.683	0.0280	0.00849	0.00588	0.00417	0.00276
10	5	7.308	6.805	5.838	5.424	4.199	0.114	0.0561	0.0210	0.00965	0.00497
10	10	8.705	8.705	8.595	7.292	4.525	0.110	0.0754	0.0450	0.0112	0.00491
10	20	4.675	4.675	4.645	3.794	1.273	0.130	0.107	0.107	0.0284	0.00507
15	10	9.855	9.811	9.208	7.378	3.929	0.379	0.281	0.158	0.0213	0.00733
20	10	11.222	11.069	10.531	5.993	3.338	0.984	0.793	0.323	0.0460	0.00973
20	20	6.609	6.609	6.498	5.460	2.323	1.027	0.907	0.539	0.084	0.0098
20	40	2.936	2.936	2.906	2.562	0.762	1.398	1.463	1.428	0.510	0.01329
40	10	9.860	9.860	9.048	7.152	2.892	8.322	6.660	2.553	0.298	0.023
40	20	6.687	6.676	6.503	5.044	1.708	11.123	10.398	5.799	0.572	0.0254
40	40	4.820	4.820	4.810	4.580	1.853	13.434	12.338	8.983	1.856	0.0350
40	80	1.950	1.950	1.950	1.670	0.638	18.276	18.295	18.280	7.280	0.0378
80	40	6.330	6.330	6.310	5.980	1.999	134.805	129.892	91.724	11.215	0.0609
80	80	2.790	2.790	2.790	2.370	0.841	235.055	229.069	198.649	50.365	0.0736
80	160	1.050	1.050	0.923	0.894	0.349	367.771	364.288	276.099	259.992	0.110

### Optimality Gaps of Heuristics.

We test the ED Metaheuristic with a threshold percentage  $p_a = 0.4$ , using the Ratio-based ED Metaheuristic, alternatively with single-denial (denoted ED 1) and all single-and-pairwise denials (denoted ED 2), as respectively defined in §3. We compare the results of these experiments utilizing the Ratio-based ED Metaheuristic variant to the optimal solutions reported by the commercial solver BARON as reported in Table 10. We utilized BARON due to its robust and efficient performance as noted in Neumaier et al. (2005), wherein the authors indicate that BARON has a 1.8% false optimal reporting rate and is designed to solve nonlinear convex optimization problems optimally.

We call attention to the negative optimality gaps within the results in Table 10.

We utilized BARON to find optimal solutions for each problem, and it reported a solution as globally optimal upon termination for each problem size’s instance where noted by an asterisk. We observe that in multiple instances, each of the solution methods is able to find superior solutions to BARON for the SWTA problem. As such, the relative gap with these instances is a negative value, and the solutions reported to be optimal by BARON are not, in fact, optimal. As a confirmatory experiment, for every instance for which an alternate solution method outperformed BARON, we conducted a warm start of BARON by initializing it with the solution attained by the ED Metaheuristic. For each such instance, BARON returned the warm start solution as optimal. For the largest problem instances, BARON was only able to find a feasible solution and a lower bound within two hours of computational effort, and these feasible solutions were inferior to those found by the QP Heuristic (7 of 20 instances), the Ratio-based ED Metaheuristic with single-denial (10 of 20 instances) and the Ratio-based ED Metaheuristic with single-and-pairwise denials (18 of 20 instances).

Table 11 reports the number of superior solutions out of 20 instances for each problem size that each solution method finds relative to the reportedly optimal solutions from BARON. We note that, for larger problem sizes having more weapons than targets, we find superior solutions to BARON with greater frequency than for problem sizes having more targets than weapons. We observe that, compared to the Ratio-based ED Metaheuristic with single-and-pairwise denials, we find 59 instances, a 21.4% false reporting rate, higher than reported by Neumaier et al. (2005).

To garner better insight regarding the performance of each heuristic, we present the relative gaps using the best solution for each instance regardless of solution technique. Table 12 depicts these relative gaps. We note that the Ratio-based ED Metaheuristic with single-and-pairwise denials is consistently within 2% of the best ob-



**Table 10. Relative Optimality Gaps with BARON (%)**

Weapons	Targets	Search Heuristic											
		QP			ED 1			ED 2			Construction		
5	5	7.299	±	8.784	4.189	±	7.576	0.467	±	1.636	0.460	±	2.059
10	5	9.098	±	9.898	4.001	±	4.169	0.834	±	1.483	1.780	±	3.501
10	10	10.793	±	8.920	5.500	±	6.606	0.596	±	1.181	3.136	±	3.430
10	20	6.501	±	3.436	4.577	±	2.341	1.447	±	1.478	6.303	±	4.525
15	10	13.138	±	9.474	8.086	±	6.876	1.491	±	1.553	5.106	±	4.012
20	10	14.148	±	8.288	9.534	±	8.778	1.082	±	2.069	1.524	±	3.485
20	20	8.857	±	3.495	6.287	±	3.208	1.728	±	1.295	2.978	±	2.185
20	40	3.916	±	1.153	2.707	±	1.376	0.855	±	0.502	5.360	±	2.153
40	10	-60.701	±	11.568	-61.802	±	11.682	-64.530	±	10.465	-64.855	±	10.141
40	20	0.083	±	10.747	-1.777	±	9.545	-6.510	±	8.679	-5.792	±	7.578
40	40	7.137	±	3.468	5.074	±	1.864	1.883	±	0.409	2.921	±	0.954
40	80	2.734	±	0.877	1.969	±	0.726	0.730	±	0.418	3.812	±	1.652
80	40	-15.719	±	12.564	-17.471	±	12.028	-21.060	±	11.543	-21.446	±	11.607
80	80	4.295	±	1.696	3.403	±	1.099	1.396	±	0.680	1.557	±	0.612
80*	160*	-0.00450	±	1.150	-0.391	±	1.146	-1.058	±	1.142	3.619	±	1.718

\*BARON did not converge to a global optimal solution within 2 hours.

served solution.

### 3.5 Conclusion

We refer to work by Kline (2017) in defining a Modified Quiz Problem (MQP) Heuristic capable of finding quality solutions to the Static Weapon Target Assignment (SWTA) Problem in thousandths of a second as a basis for an improving metaheuristic. We present the Eminent Domain (ED) Metaheuristic as a solution methodology to exploit the highly efficient MQP as a subroutine capable of finding improved solutions in real time. We remove the previously developed modification to the MQP Heuristic so as to increase its computational efficiency due to a similar process in the ED Metaheuristic to overcome a shortcoming inherent in a greedy selection process. Using the Quiz Problem (QP) heuristic as a subroutine, we demonstrate the real time improvements of the ED Metaheuristic with respect to solution quality, which often

**Table 11. Number of Solutions (out of 20 Instances per Problem Size) That Each Heuristic Solution is Superior to BARON**

Weapons	Targets	Search Heuristic			
		QP	ED 1	ED 2	Construction
5	5	0	0	0	0
10	5	0	0	0	0
10	10	0	0	0	1
10	20	0	0	0	0
15	10	0	0	1	0
20	10	0	1	5	3
20	20	0	0	0	0
20	40	0	0	0	0
40	10	20	20	20	20
40	20	7	8	12	15
40	40	0	0	0	0
40	80	0	0	0	0
80	40	19	20	20	20
80	80	0	0	1	0
80*	160*	7	10	18	1

\*BARON did not converge to a global optimal solution within 2 hours.

**Table 12. Average Relative Gaps for Selected Heuristics with Respect to the Best Solution Available (%)**

Weapons	Targets	Search Heuristic											
		QP		ED 1		ED 2		Construction					
5	5	7.300	±	8.784	4.190	±	7.576	0.468	±	1.636	0.461	±	2.058
10	5	9.099	±	9.898	4.002	±	4.169	0.835	±	1.483	1.781	±	3.501
10	10	10.794	±	8.920	5.501	±	6.607	0.597	±	1.180	3.138	±	3.429
10	20	6.501	±	3.436	4.577	±	2.341	1.447	±	1.478	6.303	±	4.525
15	10	13.153	±	9.460	8.101	±	6.861	1.504	±	1.537	5.121	±	4.021
20	10	14.828	±	8.039	10.173	±	8.377	1.702	±	2.123	2.118	±	2.573
20	20	8.857	±	3.495	6.287	±	3.208	1.728	±	1.295	2.978	±	2.185
20	40	3.916	±	1.153	2.707	±	1.376	0.855	±	0.502	5.360	±	2.153
40	10	12.702	±	6.374	9.253	±	4.389	1.679	±	1.747	0.997	±	2.105
40	20	8.065	±	3.891	6.150	±	3.273	1.043	±	1.712	1.953	±	2.427
40	40	7.137	±	3.468	5.074	±	1.864	1.883	±	0.409	2.921	±	0.954
40	80	2.734	±	0.877	1.969	±	0.726	0.730	±	0.418	3.812	±	1.652
80	40	7.732	±	3.110	5.538	±	2.324	0.910	±	1.110	0.396	±	0.604
80	80	4.301	±	1.695	3.409	±	1.102	1.401	±	0.667	1.562	±	0.613
80	160	1.087	±	0.313	0.697	±	0.249	0.022	±	0.076	4.748	±	1.105

exceed those of the best known solutions to this problem found in the literature.

In observing the relative improvement of each assignment quality calculation method and the number of dominant solutions found by these methods when compared to the QP heuristic and the Construction heuristic, we demonstrate the capability of the ED Metaheuristic to find, in real time, high quality solutions to the SWTA. We demonstrate that the ED Metaheuristic contains three parameters (i.e. the type of assignment quality computation, the number of simultaneous denials, and the threshold percent  $p_a$ ), that can be tuned to enable the Metaheuristic to effectively and efficiently solve large problem instances.

We further note that the type of assignment quality computation utilized within the ED Metaheuristic, either a Slope-based or Ratio-based method, has strengths with differing problem sizes. The ED Metaheuristic can, for smaller sized problems (less than 20 weapons and 20 targets) find better solutions using the Ratio-based Method whereas, for larger sized problems (more than 40 weapons and 10 targets) find better solutions with the Slope-based Method. However, we note that these results are simply due to the size of the set of denials,  $\mathbf{x}^d$ , for each problem size. That is, for smaller problem sizes, the Ratio-based Method has a larger set of denials whereas, for larger problem sizes, the Slope-based Method has a larger set of denials for equivalent threshold percentages.

We identify in §4 that the number of simultaneous denials affects the solution improvement for up to two simultaneous denials. Beyond this, we observe no improvements to the solutions. Whereas modifications to the ED Metaheuristic can to find improvements by using more than two simultaneous denials, we caveat that such methods would make the ED Metaheuristic more closely resemble a brute force algorithm and would quickly cease to be a real-time solution method for this problem. Moreover, we observe that the average optimality gap for the single-and-pairwise de-

nial variant of the ED Metaheuristic is exceeded by the baseline Construction Heuristic only three times but at a noteworthy computational cost. Conversely, the optimality gap of the single-denial variant of the ED Metaheuristic is often greater than the Construction Heuristic but is capable of finding these quality solutions within two tenths of a second at its slowest.

We present computational results for our testing of differing threshold percentages and note that, though more computationally expensive, setting  $p_a$  to 0% generally results in much improved solutions compared with higher  $p_a$  values. When only allowing one denial, this difference in computational time is nearly negligible, whereas for two simultaneous denials it becomes cumbersome and the average ED Metaheuristic required solution time increases notably.

We observe that the parameters allow for real time solutions which improve upon the solution found using the QP Heuristic. The single-denial variant of the ED Metaheuristic finds real-time solutions that are within 10% (on average 4%) of BARON's reported optimal solutions. Moreover, the single-and-pairwise denials ED is capable of a more rigorous search, finding solutions that are at most within 2% (on average 0.8%) of the optimum and are capable of finding solutions 64%, as reported by BARON, should there be no real-time requirement for the solution.

## IV. Transforming the Weapon Target Assignment Problem for Efficient Optimal Convergence

### 4.1 Introduction

The Weapon Target Assignment (WTA) Problem has been well studied since its introduction by Manne (1958). Two distinct variants of the problem distinguish between solving an immediate and isolated event (i.e., the Static WTA or SWTA), and solving a problem which allows for subsequent engagements (i.e., the Dynamic WTA or DWTA). The research herein addresses the SWTA.

Given a set  $J = \{1, \dots, n\}$  of incoming missiles (targets), the SWTA assigns a set  $I = \{1, \dots, m\}$  of available interceptors (weapons) to the targets, minimizing the probability of a *leaker* (i.e, a target passing through defenses). Each target has a value  $V_j$ , which quantifies its worth as a measure of lethality if not destroyed by a weapon. For each weapon type  $i$ , of which we have  $w_i$  weapons, there is a probability  $p_{ij}$  with which it will successfully destroy target  $j$ , known as the probability of kill, and  $q_{ij} = 1 - p_{ij}$  likewise indicates the probability that a single weapon of type  $i$  is unsuccessful when assigned to destroy a target  $j$ . The math programming formulation for the SWTA is:

$$\begin{aligned} \mathbf{P1} : \quad & \min_x \sum_{j \in J} V_j \prod_{i \in I} q_{ij}^{x_{ij}} \\ \text{s.t.} \quad & \sum_{j \in J} x_{ij} \leq w_i, \quad \forall i \in I, \\ & x_{ij} \in \mathbb{Z}_+, \quad \forall i \in I, j \in J, \end{aligned}$$

where  $x_{ij}$  is the number of weapons of type  $i$  assigned to target  $j$ .

Although the SWTA has been solved to optimality for smaller instances with exact algorithms in the literature, Lloyd & Witsenhausen (1986) showed the SWTA to

be NP-hard. As such, the majority of solution techniques seek to find near-optimal solutions in real time, or “fast enough to provide an engagement solution before the oncoming targets reached their goals” (Leboucher et al., 2013), to be of practical value when addressing larger-sized instances. The efficient identification of optimal solutions remains important, whether to reduce computational effort when benchmarking heuristic solution techniques for smaller instances or to expand the size of instances for which optimal solutions can be attained.

Although contemporary commercial solvers can identify optimal solutions to instances of these nonlinear integer programming formulations, a thorough understanding of the underlying principles of optimization may serve to increase the efficiency of commercial solvers via an objective function transformation within the SWTA formulation. Moreover, Kline et al. (2017) showed that commercial solvers designed for global optimization may report suboptimal solutions as global optimal solutions, and an objective function transformation within the SWTA may mitigate this shortcoming.

Alternative formulations and transformations have been explored for the SWTA, and we refer the reader to work by Kline et al. (2018) for a thorough review of the related SWTA literature. Of relevance to this study, Kwon et al. (1999) applied a linear transformation within an efficient Lagrangian relaxation solution technique. Ahuja et al. (2007) performed a logarithmic transformation for which the authors developed a linear approximation, embedded within an efficient branch and bound algorithm. Other authors have adopted simplifying assumptions to enable a more identification of optimal solutions; Murphey (2000) assumed  $p_{ij}$  to be constant for all  $i \in I$ ,  $j \in J$ , but such an approach loses problem granularity and the identified optimal solution is not guaranteed to be optimal to the original SWTA formulation..

With respect to both the required computational effort and solution quality at-

tained by the commercial solver BARON, this paper proposes and tests a reformulation of the SWTA, both without and with simple bounds on the intermediate decision variables required for the transformation, as well as with tighter lower bounds that reduce the feasible region but do not eliminate an optimal solution. In §2, we introduce a transformation and offer alternative bound tightening constraints to reduce the feasible region of the transformed SWTA problem formulation. We also prove that the introduction of a tighter constraint to intermediate decision variables within the transformed SWTA formulation notably reduces the feasible region without eliminating an optimal solution. In §3, we test the performance of a leading commercial solver designed for global optimization, when solving a set of test instances of the alternative SWTA formulations, with respect to both solution quality and required computational effort. We conclude this paper in §4 and discuss the impact of the proposed transformations to the SWTA.

## 4.2 WTA Reformulations

The basic SWTA transformation we consider is identical to that developed by Ahuja et al. (2007). We introduce an intermediate decision variable,  $z_j$ , to attain a separable, convex objective function that no longer contains the product summation operator. Although Ahuja et al. (2007) proceeded to solve such a problem as a minimum cost network flow problem, we retain the transformation as Problem **P2**



for direct solution via a commercial solver in testing herein.

$$\begin{aligned}
\mathbf{P2} : \quad & \min_{x,z} \sum_{j \in J} V_j e^{z_j} \\
\text{s.t.} \quad & \sum_{j \in J} x_{ij} \leq w_i, \quad \forall i \in I, \\
& z_j = \sum_{i \in I} x_{ij} \ln(q_{ij}), \quad j \in J, \\
& x_{ij} \in \mathbb{Z}_+, \quad \forall i \in I, j \in J.
\end{aligned}$$

Given that the BARON commercial solver leverages a branch and reduce solution procedure (Tawarmalani & Sahinidis, 2004), the imposition of finite upper and lower bounds on the intermediate decision variables,  $z_j$ ,  $\forall j \in J$ , should improve solver performance. As such, we offer Problem **P2** with additionally derived bounds as Problem **P3**, where the bounds do not reduce the feasible region of the original SWTA formulation.

$$\begin{aligned}
\mathbf{P3} : \quad & \min_{x,z} \sum_{j \in J} V_j e^{z_j} \\
\text{s.t.} \quad & \sum_{j \in J} x_{ij} \leq w_i, \quad \forall i \in I, \\
& z_j = \sum_{i \in I} x_{ij} \ln(q_{ij}), \quad j \in J, \\
& z_j \geq \sum_{i \in I} \ln(q_{ij}), \quad j \in J, \\
& z_j \leq 0, \quad j \in J, \\
& x_{ij} \in \mathbb{Z}_+, \quad \forall i \in I, j \in J.
\end{aligned}$$

We contend that, when the additional bounding constraints of **P3** are active, a

commercial solver will converge to an optimal solution more efficiently than it will for **P2**. However, we propose that, in general, the constraints in **P3** are not active and that any decrease in the required computational effort to attain a global optimal solution will not be statistically significant. Alternatively, we consider the formulation for Problem **P4**, wherein tighter bounds are emplaced upon the intermediate decision variables, and the feasible region is reduced, but an optimal solution is not eliminated by the tighter bounds.

$$\begin{aligned}
\mathbf{P4} : \quad & \min_{x,z} \sum_{j \in J} V_j e^{z_j} \\
\text{s.t.} \quad & \sum_{j \in J} x_{ij} \leq w_i, \quad \forall i \in I, \\
& z_j = \sum_{i \in I} x_{ij} \ln(q_{ij}), \quad j \in J, \\
& z_j \geq c \min_{i \in I} \{\ln(q_{ij})\}, \quad j \in J, \\
& z_j \leq 0, \quad j \in J, \\
& x_{ij} \in \mathbb{Z}_+, \quad \forall i \in I, j \in J,
\end{aligned}$$

wherein the parameter  $c$  is instance-specific, and it characterizes the possible difference in the number of weapons assigned to any one target from the number assigned to any other target in an optimal solution.

**Lemma 1** *Any instance of the SWTA in which there exists an  $i \in I$  for which  $w_i > 1$  can alternatively be expressed as an instance in which  $w_i = 1$  for all  $i \in I$ .*

PROOF. Given an instance of Problem **P1** having a set  $I$  of weapon types, each with  $w_i$  of each type and probability of intercept  $p_{ij}$  against target  $j \in J$ . This yields a total of  $\sum_{i \in I} w_i$  individual weapons in the original instance. For the reformulated instance, let  $|I'| \leftarrow \sum_{i \in I} w_i$ ;  $w_{i'} \leftarrow 1$ ,  $\forall i' \in I'$ ; and  $p_{i'j} \leftarrow p_{ij}$  for exactly  $w_i$  weapon

types  $i' \in I'$  for each  $i \in I$ . Every feasible solution for the original instance has a corresponding solution in the reformulated instance, wherein multiple assignments of a single weapon type in the original instance are distributed among identical weapon types in the reformulated instance.  $\square$

The following theorem addresses the calculation of  $c$ .

**Theorem 2** *Given an instance of the SWTA having a set of targets  $J = \{1, \dots, n\}$  and a set of weapons  $I = \{1, \dots, m\}$  with  $w_i = 1$  for all  $i \in I$ , the difference between the number of weapons that are assigned to target 1 and the number of weapons that can be assigned to target  $n$  in an optimal solution will not exceed  $c$ , where  $c$  is given by*

$$c = \max_{k \in \mathbb{Z}} \{V_1(1 - p_{i1})^k > V_n\} + 1.$$

PROOF. There are three cases to consider:  $|I| = |J|$ ,  $|I| > |J|$ , and  $|I| < |J|$ . For each case, the probability of kill is randomly chosen such that  $p^{\min} \leq p_{ij} \leq p^{\max}$ . Because the maximum number of weapons that can be assigned to any one target will occur at lower probabilities of kill, let  $p = p^{\min}$  and  $q = 1 - p$ . Further, for each case, let  $V_1 \geq V_2 \geq \dots \geq V_n$  without loss of generality (w.l.o.g.).

Case 1:  $|I| = |J|$ . Compare the objective functions of a solution having one weapon assigned to each of the targets with a solution having at least two weapons assigned to one target. Given the possible values of  $V_j$  and  $p_{ij}$ , two weapons can be assigned to Target 1 and zero weapons assigned to Target  $n$  in an optimal solution if

the following inequality is feasible:

$$V_1q^2 + V_2q + \cdots + V_{n-1}q + V_n < V_1q + V_2q + \cdots + V_{n-1}q + V_nq$$

$$V_1q^2 + V_n < V_1q + V_nq$$

$$V_1q(1 - q) > V_n(1 - q)$$

$$V_1q > V_n$$

Given  $V_1q > V_n$  is possible, three weapons can be assigned to Target 1 and zero assigned to Targets  $n$  and  $n - 1$  in an optimal solution if the following inequality is feasible.

$$V_1q^3 + V_2q + \cdots + V_{n-1} + V_n < V_1q^2 + V_2q + \cdots + V_{n-1}q + V_n$$

$$V_1q^3 + V_{n-1} < V_1q^2 + V_{n-1}$$

$$V_1q^2(1 - q) > V_{n-1}(1 - q)$$

$$V_1q^2 > V_{n-1}$$

This relationship continues until all weapons are assigned to Target 1 and Targets  $2, \dots, n$  have zero weapons assigned. Therefore, when  $V_1(1 - p)^k > V_2$  is feasible,  $c = k + 1$  weapons can be assigned to Target 1 in an optimal solution, leaving  $k$  targets unassigned.

Case 2:  $|I| < |J|$ . At least  $|J| - |I| = s$  targets will not have a weapon assigned and two weapons can be assigned to Target 1 in an optimal solution if the following

inequality is feasible.

$$\begin{aligned}
V_1q^2 + V_2q + \cdots + V_{n-s+1} + V_{n-s} + \cdots + V_n &< \\
V_1q + V_2q + \cdots + V_{n-s+1}q + V_{n-s} + \cdots + V_n & \\
V_1q^2 + V_{n-s+1} &< V_1q + V_{n-s+1}q \\
V_1q(1-q) &> V_{n-s+1}(1-q) \\
V_1q &> V_{n-s+1}
\end{aligned}$$

The same relationship holds for Case 2 as for Case 1. Therefore, when  $V_1(1-p)^k > V_{n-s+1}$  is possible,  $c = k + 1$  weapons can be assigned to one target in an optimal solution, leaving  $s + k$  targets unassigned.

Case 3:  $|I| > |J|$ . At least  $|J| - |I| = s$  targets will have at least  $r = \left\lfloor \frac{|I|}{|J|} \right\rfloor$  assigned weapons with  $r + 1$  assigned to the remaining targets. However, an optimal solution may have  $r + 2$  weapons assigned to Target 1 and  $r$  assigned to Target  $s - 1$  when the following inequality is feasible.

$$\begin{aligned}
V_1q^{r+2} + V_2q^{r+1} + \cdots + V_{n-s+1}q^r + V_{n-s}q^r + \cdots + V_nq^r &< \\
V_1q^{r+1} + V_2q^{r+1} + \cdots + V_{n-s+1}q^{r+1} + V_{n-s}q^r + \cdots + V_nq^r & \\
V_1q^2 + V_{n-s+1} &< V_1q + V_{n-s+1}q \\
V_1q(1-q) &> V_{n-s+1}(1-q) \\
V_1q &> V_{n-s+1}
\end{aligned}$$

The same relationship holds for Case 3 as for Case 1. Therefore, an optimal solution may exist that has  $c = k + 2$  more weapons assigned to Target 1 than assigned to Targets  $n - s + 1$  through  $n$  when  $V_1(1-p)^k > V_{s-1}$  is possible.  $\square$

### 4.3 Testing and Results

We conduct an experiments to examine whether formulations **P2-P4** improve upon **P1** with respect to solver performance. This experiments uses the GAMS Algebraic Modeling Language, version 24.8.5, with the commercial solver BARON, version 17.4.1. The experiment considers the problem instance sizes shown in Table 13. We define parameters  $V_j$  and  $p_{ij}$  as continuous uniformly distributed random variables  $V_j \in [25, 100]$  and  $p_{ij} \in [0.6, 0.9]$ . We solve 30 instances of each problem size for each of the four problem formulations. We perform all tests on a computer having an Intel Xeon E5-2650 v2 processor with 128 GB RAM.

**Table 13. Tested Problem Instance Sizes**

Problem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Weapons	5	10	10	10	15	20	20	20	40	40	40	40	80	80	80
Targets	5	5	10	20	10	10	20	40	10	20	40	80	40	80	160

Given the distributions of our parameters, we note that

$$V_1 q > V_n$$

$$100(0.4) > 25$$

$$40 > 25$$

is possible, thus we can have two more weapons assigned to Target 1 than to Target  $n$  in an optimal solution. However

$$V_1 q^2 > V_n$$

$$100(0.4)^2 > 25$$

$$16 > 25$$

cannot hold, so an optimal solution cannot have two more weapons assigned to Target 1 than Target  $n$ . Therefore, within **P4**, we set  $c = 2$  when  $|I| \leq |J|$  and  $c = \left\lfloor \frac{|I|}{|J|} \right\rfloor + 3$  when  $|I| > |J|$ .

Table 14 reports the experimental results of required computational effort for BARON to find an optimal solution for each of **P1-P4** and the results of the two-sample  $t$ -test having the null hypotheses that the required solution time for **P1** is less than that required for **P2**, the required solution time for **P2** is less than that required for **P3**, and the required solution time for **P3** is less than that required for **P4**. We define our significance level to be  $\alpha = 0.5$ . Additionally, we note that BARON required lesser computational effort to solve Problem **P4**, on average, for 11 out of 15 (i.e., 73%) problem instance sizes when compared to Problems **P3** and **P2**, and 9 out of 15 (i.e., 60%) problem instance sizes when compared to Problem **P1**.

**Table 14. Average Solution Times and Hypothesis Tests**

W	T	Mean required computational effort (sec)				$H_0$ (required computational effort)		
		<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	<b>P1 &lt; P2</b>	<b>P2 &lt; P3</b>	<b>P3 &lt; P4</b>
5	5	0.0287	0.0287	0.0290	0.0553	R	R	FTR
10	5	0.161	0.0680	0.0667	0.0943	R	R	FTR
10	10	0.365	0.208	0.177	0.243	R	R	FTR
10	20	3.547	98.450	97.548	2.256	FTR	R	R
15	10	1.085	10.410	16.824	5.772	FTR	R	R
20	10	2.179	0.504	0.539	0.384	R	R	R
20	20	1.550	4.062	4.407	0.538	FTR	R	R
20	40	34.552	1800.002	1800.001	32.749	FTR	R	R
40	10	15.021	968.814	978.820	310.017	FTR	R	R
40	20	25.836	415.600	430.382	77.990	FTR	R	R
40	40	49.616	1292.189	1296.902	36.032	FTR	R	R
40	80	94.731	1800.001	1800.002	510.491	FTR	R	R
80	40	350.690	1800.000	1800.002	1800.001	FTR	R	R
80	80	570.158	1800.001	1800.002	521.411	FTR	R	R
80	160	1804.966	1801.095	1801.102	1797.776	R	R	R

We observe in Table 14 that mean solution times support our conjecture that the reformulation with bound tightening constraints will increase the efficiency of the

commercial solver, BARON. For any of the smaller problems (i.e., Problem sizes 1 - 7), we note that the mean required computational efforts for problems for which the ratio of the number of weapons to the number of targets is integer-valued (i.e., Problem sizes 2 and 6) is much lower for **P2** and **P3** than for **P1**. We caveat that the mean solution time for Problem sizes 4 and 5, for which the weapon-to-target ratios are  $\frac{1}{2}$  and  $\frac{3}{2}$ , respectively, are much greater for **P2** and **P3** than for **P1**. Further, we note the relatively large increase in required computational effort for Problems 8-15 which we attribute to the combinatorial explosion associated with this NP-hard problem.

We also note that the hypothesis testing in Table 14 shows that only in Problems 1, 2, 3, and 6 does no statistical difference in convergence speed exist when comparing the solver performance for **P1** and **P2**. There is no statistical decrease in required computational effort between **P2** and **P3**. For Problems 4-15, we can reject the null hypothesis that **P3** will converge faster than **P4**, and we can therefore state that the convergence rates of **P4** are less than or equal to those of **P3**. From the results reported in Table 14, it is evident that **P4** requires less time for BARON to identify a global optimum than either **P2** or **P3**. We also notice that, in examining the convergence rates for **P1** and **P4**, no single formulation yields a performance by BARON that consistently dominates another formulation.

The other conjecture states that the solutions found by solving **P2** and **P3** will be no worse than **P1**. While this may seem an odd statement when comparing results of a global optimization tool, we find that BARON reports suboptimal solutions when solving **P1**. Table 15 shows that, as the size of the problem increases, **P1** becomes less reliable, especially in cases where there are more weapons than targets. We attribute this to the functional approximation to the polyhedral outer-approximation, which we believe to be a loose approximation for **P1** but a tight approximation for **P2** and



**P3** due to the separable convex functions of the latter. We find that the solutions identified by solving **P4** are identical to those found by **P2** and **P3** when the latter converge to an optimal solution within 30 minutes. When **P2** and **P3** do not converge to an optimal solution within 30 minutes, **P4** improves upon the best solution by 0.098%, on average.

**Table 15. P1 Suboptimal Solutions**

w	t	Mean relative gap (%)	Number suboptimal
5	5	0	0
10	5	0	0
10	10	0	0
10	20	0	0
15	10	0.02	2
20	10	0.13	8
20	20	0	0
20	40	0	0
40	10	9.61	28
40	20	6.91	29
40	40	0.01	6
40	80	0	0
80	40	11.7	29
80	80	5.04	29
80	160	7.91	30

#### 4.4 Conclusion

Global optimization is a challenging task made easier by powerful commercial solvers. Among available commercial solvers, BARON is regarded as both efficient and robust (Neumaier et al., 2005). Yet, it is clear that even one of the leading global optimization solvers can converge to suboptimal solutions or expend unnecessary computational time if the formulation is ill-suited for the solver.

Herein we defined a transformation on the Static Weapon Target Assignment (SWTA) Problem and offered several alternative sets of bound tightening constraints

which reduced the solution space while preserving the global optimal solution given the specific parameters of this problem. We hypothesized that BARON would solve **P4** more efficiently than **P2** and **P3**, and that the solutions for **P2**, **P3**, and **P4** would be non-degrading as compared to **P1**. We empirically demonstrated that the transformed problem formulations did increase the efficiency of finding an optimal solution for smaller problems under certain conditions, and that it increased solution quality up to 11.7% of solutions using the untransformed problem formulation. We used two-sample *t*-tests to test for statistical significance to the hypotheses that one formulation performs more or less efficiently than another, and results aligned with our hypotheses and conjecture.

Ultimately, this exploration into the performance of a commercial solver for different bounding techniques on a logarithmic transformation serves to inform our ability to efficiently and effectively use a global optimization solver. Although commercial solvers are continuously improving, an understanding of how these solvers function enables a user to provide the best formulation, lest the solution be suboptimal and/or computationally expensive.

## V. A Greedy Hungarian-like Algorithm for the SWTA

### 5.1 Introduction

The air defense problem of assigning available interceptors (weapons) to engage incoming missiles (targets) is known in the literature as the Weapon Target Assignment (WTA) Problem. The WTA Problem was introduced to the field by Manne, who derived of it from a presentation by Merrill Flood in 1957 at The Princeton University Conference on Linear Programming Manne (1958). Variants of the problem as defined first by Matlin (1970) include the Static WTA (SWTA) and Dynamic WTA (DWTA). This paper focuses on the SWTA Problem and obtaining quality solutions to this NP-Hard problem Lloyd & Witsenhausen (1986) in real-time or “fast enough to provide an engagement solution before the oncoming targets reach their goals” Leboucher et al. (2013).

Given  $n$  incoming targets, solving this defensive variant of the problem results in the assignment of  $m$  weapon types to engage the targets so as to minimize the collective expected residual value of the targets. The value of target  $j$ ,  $V_j$ , corresponds to its destructive capacity and weapons of type  $i$ , and there are  $w_i$  such weapons, have an associated probability  $p_{ij}$  of destroying target  $j$ . The problem seeks to minimize the residual value of each target, known in the literature as target leakage, which utilizes the probability of survival, defined as  $q_{ij} = 1 - p_{ij}$ . The SWTA problem is nonlinear and is defined by

$$\begin{aligned} \min \quad & \sum_{j=1}^n V_j \prod_{i=1}^m q_{ij}^{x_{ij}} \\ \text{st} \quad & \sum_{j=1}^n x_{ij} \leq w_i, \text{ for } i = 1, \dots, m, \\ & x_{ij} \in \mathbb{Z}_+, \text{ for } i = 1, \dots, m, \ j = 1, \dots, n \end{aligned} \tag{1}$$

where the decision variable  $x_{ij}$  is the number of weapons of type  $i$  to assign to target  $j$ . In this paper, we address the most general case where  $w_i = 1$  for  $i = 1, \dots, m$ , which results in a binary decision variable.

Since its inception, many solution techniques have been applied to variants of the problem, from exact algorithms to heuristic and metaheuristic algorithms. Kline (2017) proved the convexity of the untransformed SWTA Problem in 2017 and developed a method of finding exact solutions to smaller problem sizes. Others utilize transformations in order to expedite optimal algorithms, such as Ahuja et al. (2007), who transforms the objective function and uses a branch and bound algorithm, and Malcolm (2004), who transforms the constraint matrix, simplifies the objective function, and uses the simplex method. Many well known heuristic algorithms have been utilized to solve the SWTA, such as the Very Large Scale Neighborhood (VLSN) search Lee (2010) and the genetic algorithm Lee et al. (2003), Bogdanowicz et al. (2013).

In this paper, we review the Quiz Problem (QP) Heuristic and the Eminent Domain (ED) Metaheuristic from Kline et al. (2017) in §2. In §3, we introduce a new heuristic which shares some similarities with the Hungarian Algorithm. We test the performance of this heuristic and compare it to the QP heuristic with and without the ED Metaheuristic and the lower bounds as reported by BARON in §4 and identify our conclusions in §5.

## 5.2 QP Heuristic and ED Metaheuristic

### QP Heuristic.

Kline (2017) proposed application of the optimal solution to the quiz problem to a heuristic algorithm for the weapon target assignment problem. Bertsekas & Castañon (1999) first applied this methodology to a heuristic search for scheduling

problems and later Ahner (2005) used the approach to route unmanned aerial vehicles in risky environments. The quiz problem states that an individual presented with a series of questions with values  $v_u$  has probability  $p_u$  of correctly answering a question, where  $u = 1, \dots, n$ . Further, the individual answers questions, receiving the value of each question as a reward if correct, until he answers one incorrectly, at which point the quiz is terminated. The goal of the quiz problem is to identify the order in which to answer the questions to maximize the sum value of those correctly answered. Bertsekas & Castañón (1999) showed that the strategy to maximize return is an index policy in which questions are answered in descending values of  $y_u$ , where

$$y_u = \frac{v_u p_u}{1 - p_u} = \frac{v_u p_u}{q_u}.$$

For the SWTA, we use the quiz problem strategy to define the value of each weapon-target assignment as  $y_{ij}^0$ , which allows us to select the maximum return for a single weapon-target assignment  $x_{\hat{i}\hat{j}}$ , where

$$(\hat{i}\hat{j}) \in \arg \max_{\substack{i=1,\dots,m \\ j=1,\dots,n}} \{y_{ij}^0\}.$$

We then redefine our target value  $V_j = V_j q_{\hat{i}\hat{j}}$ , which is the residual value of the selected target given the weapon assigned. If there remains only one available  $w_i$ , we redefine our probabilities of kill for this weapon  $p_i = 0$ . Using these updated values, we define the next iteration value array as  $y_{ij}^1$ . We repeat this process until each of the weapons is assigned to a target.

### **ED Metaheuristic.**

Kline et al. (2017) developed the Eminent Domain (ED) Metaheuristic as an efficient method to improve upon a feasible solution by iteratively preventing assignments

which may result in suboptimal solutions. Utilizing the QP Heuristic as a subroutine, the ED Metaheuristic is initialized by finding a feasible solution and assessing the fitness of each assignment. From this, the ED Metaheuristic iteratively blocks each of a subset of the assignments in the initial solution based upon this fitness metric, either individual blocks (the ED 1 Metaheuristic) or individual-and-pairwise blocks (the ED 2 Metaheuristic) using the QP heuristic while not allowing the blocked assignment(s). If a superior solution is found, the process is repeated with the improved solution until no improvements are found.

The ED Metaheuristic may find superior solutions because greedy selection processes sequentially make assignments based upon the maximum immediate utility of all available assignments which can make assignments whose selection result in inferior solutions. That is, the collective fitness of the most beneficial assignment and the subsequent assignments is inferior to a solution having a less superior first assignment and subsequent assignments. As this is a common characteristic of greedy search heuristics, and of heuristics in general, the ED Metaheuristic prevents each of the potentially obstructive assignments to allow for the possibility of a superior collection of assignments to be made. The ED Metaheuristic results in monotonically improving solutions.

### 5.3 Greedy Hungarian Heuristic

In examining the optimal solutions to smaller problems and contrasting suboptimal solutions using iterative selection processes described heretofore, we observe a property common to optimal solutions: optimal assignments tend to be among the best available for each weapon and for each target; local optimal assignments may be elements of a global optimal solution. We note that, for each weapon, the expected value of survival,  $V_j(1 - p_{ij})$  or  $V_j q_{ij}$ , of each target decays at inconsistent rates. For

one weapon, the second best assignment available may be marginally smaller than the best available assignment whereas a different weapon may have far greater decay. A different decay is observable when examining the assignments available to each target. Those assignments which are part of a globally optimal solution generally do not suffer much decay from the best assignments available to each weapon or each target.

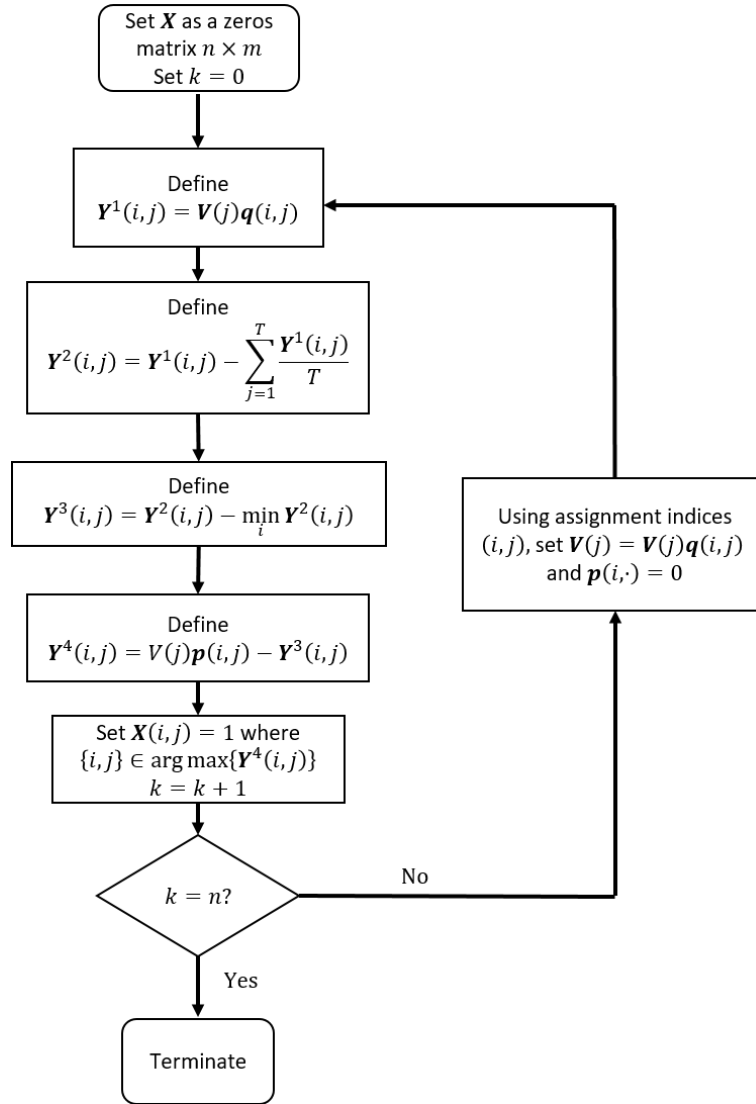


Figure 5. Greedy Hungarian Heuristic Flow Chart

We develop a Greedy Hungarian-like Heuristic, which develops a relative fitness

for each assignment, illustrated in Figure 5. We initialize this process by defining an assignment matrix of zeros,  $\mathbf{X}$ . We define a matrix,  $\mathbf{Y}^1$ , of the expected value of survival for each assignment,  $V_j q_{ij}$ . We next seek to scale down the assignment values in order to assess their fitness relative to one another without artificially inflating those of targets with larger values. We do this by subtracting from  $\mathbf{Y}^1$  a matrix of the mean column values of  $\mathbf{Y}^1$ . This results in a new matrix  $\mathbf{Y}^2$ , whose columns have a mean of 0. We next define a matrix,  $\mathbf{Y}^3$ , as the difference of  $\mathbf{Y}^2$  and a matrix of the minimum values of each row of  $\mathbf{Y}^2$ . This matrix,  $\mathbf{Y}^3$ , has a minimum value of 0 where the best assignment for each weapon is met by an assignment among the best for each target. Next, we define  $\mathbf{Y}^4$  as a matrix of the expected value of destruction,  $V_j p_{ij}$ , for each assignment less  $\mathbf{Y}^3$ . We update our assignment matrix  $\mathbf{X}$  by incrementing  $\mathbf{X}_{ij}$ , where  $\{i, j\}$  corresponds to the maximum value of  $\mathbf{Y}^4$ . If any weapons remain unassigned, we set the probabilities of kill for weapon  $i$  to 0 and redefine the value of target  $j$  as  $V_j = V_j q_{ij}$  and repeat the above process, starting with defining  $\mathbf{Y}^1$ .

We present an example of this heuristic in Table 16. Given 5 weapons and 5 targets, we first see, in matrix  $\mathbf{Y}^1$ , the expected survival of each assignment. Matrix  $\mathbf{Y}^2$  shows the next step, where the average expected survival for each target is subtracted from each assignment of that target. In matrix  $\mathbf{Y}^3$ , the minimum value for each weapon is subtracted from each value for that weapon. Lastly, we see in matrix  $\mathbf{Y}^4$  the difference of matrix  $\mathbf{Y}^3$  from the expected kill value of each assignment (denoted  $\mathbf{V}_j \mathbf{p}_{ij}$ ). We identify the maximum value of matrix  $\mathbf{Y}^4$ , which is the assignment of weapon 3 to target 2, set this value in our assignment matrix  $\mathbf{X}$ , update  $V_2$  and  $p_{3,}$ , and repeat the process until all five weapons have been assigned.

We note that this heuristic is similar to the Hungarian Algorithm defined by Kuhn (1955) prior to defining the value matrix,  $\mathbf{Y}^3$ . In contrast to the Hungarian Algorithm, we define  $\mathbf{Y}^2$  as the difference of  $\mathbf{Y}^1$  and the mean value of the columns, rather than



**Table 16. Example of Greedy Hungarian Heuristic**

$\mathbf{Y}^1$	$j = 1$	2	3	4	5
$i = 1$	6.6818	19.6212	5.375	20.1761	5.5313
2	8.4738	22.7763	4.3047	18.4437	6.8691
3	9.4819	10.0417	7.8144	12.6709	8.9564
4	10.8689	13.4611	7.1515	12.541	10.6329
5	7.6667	18.8394	7.0631	7.0537	5.1386
$\mathbf{Y}^2$	$j = 1$	2	3	4	5
$i = 1$	-1.9528	2.6733	-0.9667	5.9990	-1.8944
2	-0.1608	5.8284	-2.0370	4.2666	-0.5566
3	0.8473	-6.9062	1.4727	-1.5062	1.5307
4	2.2343	-3.4868	0.8098	-1.6361	3.2072
5	-0.9679	1.8915	0.7214	-7.1234	-2.2871
$\mathbf{Y}^3$	$j = 1$	2	3	4	5
$i = 1$	0	4.6261	0.9861	7.9518	0.0585
2	1.8762	7.8654	0	6.3037	1.4805
3	7.7535	0	8.3789	5.4001	8.4370
4	5.7211	0	4.2966	1.8508	6.6941
5	6.1555	9.0148	7.8447	0	4.8363
$\mathbf{V}_j \mathbf{p}_{ij}$	$j = 1$	2	3	4	5
$i = 1$	23.5362	69.0119	20.6657	40.1949	27.7033
2	21.7442	65.8568	21.7360	41.9273	26.3655
3	20.7361	78.5914	18.2263	47.7001	24.2782
4	19.3491	75.1721	18.8893	47.8300	22.6018
5	22.5513	69.7938	18.9777	53.3174	28.0960
$\mathbf{Y}^4$	$j = 1$	2	3	4	5
$i = 1$	23.5362	64.3859	19.6797	32.2431	27.6449
2	19.8680	57.9914	21.7360	35.6237	24.8850
3	12.9825	78.5914	9.8474	42.3001	15.8412
4	13.6280	75.1721	14.5927	45.9793	15.9077
5	16.3958	60.7789	11.1329	53.3174	23.2597

**Table 17. Example using Hungarian Algorithm**

$\mathbf{Y}^2$	$j = 1$	2	3	4	5
$i = 1$	0	9.5795	1.0703	13.1225	0.3927
2	1.7920	12.7346	0	11.3901	1.7305
3	2.8002	0	3.5097	5.6173	3.8178
4	4.1871	3.4194	2.8468	5.4874	5.4942
5	0.9850	8.7976	2.7584	0	0
$\mathbf{Y}^3$	$j = 1$	2	3	4	5
$i = 1$	0	9.5795	1.0703	13.1225	0.3927
2	1.7920	12.7346	0	11.3901	1.7305
3	2.8002	0	3.5097	5.6173	3.8178
4	1.3403	0.5726	0	2.6406	2.6475
5	0.9850	8.7976	2.7584	0	0

the minimum value of the columns. We find that utilizing the minimum column value reduces the importance paid to the fitness of assignments for each weapon and often requires many iterations before a solution can be observed whereas the proposed method requires as many iterations as the number of weapons. For example, we see in Table 17 that the values of matrix  $\mathbf{Y}^2$  contain multiple zeros in rows 4 and 5 and that the subsequent matrix  $\mathbf{Y}^3$  requires additional steps before an optimal solution can be observed using the Hungarian Algorithm. By contrast, we see in Table 16 that the values of matrix  $\mathbf{Y}^2$  contains negative values which preserves the capacity for stratification shown in matrix  $\mathbf{Y}^3$ . Though this does not guarantee optimality, it is an efficient method by which we assess the fitness of each assignment in terms of targets and weapons. Additionally, we can see in Table 17 that no solution can be identified in  $\mathbf{Y}^3$  because all of the zeros can be covered with four lines. As such, additional iterations are required before an optimal solution can be identified.

## 5.4 Computational Results

We test each heuristic and metaheuristic to solve a set of problem instances by designing random parameters within various instance sizes ranging from 5 weapons

and 5 targets to 80 weapons and 160 targets. We denote the results of the ED 2 Metaheuristic, utilizing the Quiz Problem Heuristic as a subroutine, as ED 2, and the ED 1 and ED 2 Metaheuristics, utilizing the Greedy Hungarian Heuristic as a subroutine, as GH 1 and GH 2, respectively. We consider 15 problem sizes, shown in Table 18, which fix the number of weapons and targets while randomly generating target values and probabilities of kill for weapon-target pairs. For each target, a uniformly distributed continuous variable value  $[25, 100]$  is randomly generated, and we also assigned randomly generated probabilities of kill as uniformly distributed continuous variables  $[0.6, 0.9]$  so that each weapon has a different probability of kill for each target. This allows us to compare results to the performance of the QP heuristic and the ED metaheuristic, proposed by Kline et al. (2017). We generate 20 problem instances of random numbers for each of our 15 problem sizes and performed all tests on a computer having an Intel Xeon E5-2650 v2 processor with 128 GB RAM. Each solution method is applied to the same set of 20 problem to compare solution values and computational times. Our results, insights, and analysis are presented in this section.

**Table 18. Tested Problem Sizes**

Problem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Weapons	5	10	10	10	15	20	20	20	40	40	40	40	80	80	80
Targets	5	5	10	20	10	10	20	40	10	20	40	80	40	80	160

We utilize the BARON solver for each instance of each problem in order to obtain an optimal solution to which we can compare the different solution techniques on optimality gaps. According to Neumaier et al. (2005), BARON is the most robust and efficient of available solvers and has an average false optimal reporting rate of 1.8%. We can see in Table 19 that the GH heuristic performs better than the QP heuristic in every problem and that the implementation of the ED metaheuristic

while utilizing the GH heuristic as a subroutine performs comparably to the ED 2 metaheuristic.

The negative optimality gaps indicate that these solution techniques find solutions which improve upon the reported optimal solutions of BARON as noted in Kline et al. (2017), which indicates either an error within our heuristics or a falsely reported “optimal solution” from BARON. In order to address this, we run BARON a second time for each instance wherein an improved solution is found, seeding the solver with the best solution we find using one of our solution techniques. We find that, in every instance, BARON returns our solution as the “optimal solution”. This does not lead us to believe optimal solutions are found for each of these problems, but rather that BARON is reporting these cases as locally optimal and is unable to find improved solutions.

**Table 19. Average Gap with BARON**

Weapons	Targets	Average Gap (%)				
		QP	ED 2	GH	GH 1	GH 2
5	5	7.30	0.47	3.20	1.48	0.57
10	5	9.10	0.83	7.38	2.50	1.38
10	10	10.79	0.60	5.40	2.29	1.70
10	20	6.50	1.45	1.23	0.44	0.22
15	10	13.14	1.46	8.15	3.59	2.26
20	10	14.15	1.06	6.33	3.44	1.73
20	20	8.86	1.58	7.37	4.13	2.08
20	40	3.92	0.85	0.94	0.55	0.33
40	10	-60.70	-64.64	-62.39	-63.58	-64.50
40	20	0.08	-6.74	-1.87	-4.59	-6.08
40	40	7.14	1.88	5.56	3.48	2.66
40	80	2.73	0.73	0.93	0.64	0.50
80	40	-15.72	-21.11	-18.52	-19.82	-20.81
80	80	4.29	1.37	3.21	2.49	1.86
80*	160*	0.00	-1.06	-1.10	-1.26	-1.35

\*BARON did not converge to a global optimal solution within 30 minutes.

We find that applying the logarithmic transformation, shown below, to the SWTA formulation enables BARON to converge to a superior solution. We show in Table 20 that the optimality gap using the solutions of the transformed problems demonstrate solutions approaching the best found by BARON on the transformed problem but in no instance does a heuristic technique outperform BARON.

$$\begin{aligned}
\min \quad & \sum_{j=1}^n V_j e^{z_j} \\
\text{st} \quad & \sum_{j=1}^n x_{ij} \leq w_i, \text{ for } i = 1, \dots, m, \\
& z_j = \sum_{i=1}^m x_{ij} \ln(q_{ij}), \text{ for } j = 1, \dots, n, \\
& x_{ij} \in \mathbb{Z}_+, \text{ for } i = 1, \dots, m, \text{ } j = 1, \dots, n
\end{aligned} \tag{1}$$

**Table 20. Average Optimality Gap with BARON Using Transformation**

Weapons	Targets	Average Optimality Gap (%)				
		QP	ED 2	GH	GH 1	GH 2
5	5	7.30	0.47	3.20	1.48	0.57
10	5	9.13	0.86	7.40	2.53	1.41
10	10	10.79	0.60	5.40	2.29	1.70
10	20	6.50	1.45	1.23	0.44	0.22
15	10	13.15	1.48	8.16	3.60	2.28
20	10	14.98	1.81	7.13	4.21	2.47
20	20	8.86	1.58	7.37	4.13	2.08
20	40	3.92	0.85	0.94	0.55	0.33
40	10	16.09	4.36	11.29	7.54	4.69
40	20	10.90	3.46	8.74	5.80	4.17
40	40	7.16	1.89	5.58	3.49	2.68
40	80	2.73	0.73	0.93	0.64	0.50
80	40	10.30	3.25	6.67	4.96	3.66
80	80	4.65	1.72	3.56	2.85	2.21
80*	160*	1.85	0.77	0.73	0.57	0.47

\*BARON did not converge to a global optimal solution within 30 minutes.

As shown in Table 20, the ED 2 Metaheuristic is the superlative technique in terms of solution quality, yet the computational time requirements of each solution technique must be addressed as we are seeking real-time solution techniques. BARON solution times varied from 0.5 seconds to being unable to converge for the largest problem instance considered in the 30 minutes allotted. We see in Table 21 that the GH heuristic is capable of finding solutions to even the largest problems within 0.025 seconds. Meanwhile, the GH 1 and GH 2 Metaheuristics, both finding solutions within 7.54% and 4.69% of the best found solution, respectively, require just over one second and one minute, respectively. Though the ED 2 Metaheuristic generally finds the best solution, it is the slowest solution technique and ceases to be a real time metaheuristic for problems with more than 20 weapons.

**Table 21. Average Computational Time Requirements**

Weapons	Targets	Average Computational Time (sec)				
		QP	ED 2	GH	GH 1	GH 2
5	5	0.000188	0.0142	0.000371	0.00278	0.00454
10	5	0.000340	0.0717	0.000448	0.00480	0.0256
10	10	0.000342	0.0685	0.000452	0.00493	0.0268
10	20	0.000356	0.128	0.000492	0.00519	0.0322
15	10	0.000505	0.281	0.000716	0.0108	0.0945
20	10	0.000659	0.692	0.000966	0.0193	0.214
20	20	0.000706	0.713	0.00102	0.02136	0.219
20	40	0.00134	1.43	0.00129	0.02680	0.268
40	10	0.00135	5.54	0.00201	0.08818	1.61
40	20	0.00249	8.13	0.00235	0.101	1.93
40	40	0.00325	11.5	0.00296	0.126	2.39
40	80	0.00397	18.4	0.00458	0.195	3.53
80	40	0.00761	116	0.00833	0.674	25.6
80	80	0.00880	209	0.0117	1.02	38.0
80	160	0.0128	333	0.0232	1.82	73.9

## 5.5 Conclusion

We refer to Kline (2017) in defining a Modified Quiz Problem (MQP) Heuristic and to Kline et al. (2017) in defining the Eminent Domain (ED) Metaheuristic, both capable of finding quality solutions to the Static Weapon Target Assignment Problem efficiently. We present the Greedy Hungarian (GH) Heuristic as a new technique in solving the largest problems in the literature within 0.025 seconds. Further, we apply the ED Metaheuristic using the GH Heuristic as a subroutine, resulting in improved solutions in “real-time”.

We seek to quantify the fitness of our solutions by implementing the BARON solver to find the optimal solutions from which we can report optimality gaps. We find that for roughly 20% of the instances, the developed heuristic and metaheuristic solution techniques are able to find solutions superior to those reported by BARON to be optimal. Of the remaining instances, we find the GH Heuristic has an average optimality gap of 5.22% while the GH ED 1 and GH ED 2 Metaheuristics have average optimality gaps of 3.01% and 1.96%, respectively.

We show that, while the ED 2 finds the superior solution in 44% of the instances when compared to the GH 2, which finds superior solutions to the ED 2 in 37% of the instances, the ED 2 has a higher growth of required computational effort as a function of problem size than the GH 2. Further, while the GH Heuristic requires among the least computational effort, the GH 1 increases the performance of the GH by 42.5% and requires less than 1 second of computational effort for all but the largest two problems considered. Thus, we identify the GH 1 a “real-time” solution technique capable of finding near optimal solutions to the Static Weapon Target Assignment Problem.

## VI. Implementing the CAVE Algorithm for the Heterogeneous 2 Stage DWTA

### 6.1 Introduction

The problem of assigning available interceptors to incoming missiles is known in the literature as the Weapon Target Assignment (WTA) Problem. Manne (1958) put forth the first definition of the problem, from which many variations have been defined and solved.

In its simplest form, the WTA problem is defined as follows. Given  $n$  incoming missiles (targets), the Static WTA (SWTA) seeks to employ available interceptors (weapons) so as to intercept and destroy a subset of the targets, minimizing the expected value of any *leakers*, targets passing through defenses. Each target has a value  $V_j$ , which quantifies its lethality, and for each weapon type  $i$ , of which we have  $w_i$  weapons, there is a probability  $p_{ij}$  with which it will successfully destroy target  $j$ , known as the probability of kill. The SWTA formulation is

$$\begin{aligned} \min \quad & \sum_{j=1}^n V_j \prod_{i=1}^m (1 - p_{ij})^{x_{ij}} \\ \text{st} \quad & \sum_{j=1}^n x_{ij} \leq w_i \text{ for } i = 1, \dots, m, \\ & x_{ij} \in \mathbb{Z}_+, \text{ for } i = \{1, \dots, m\}, \quad j = \{1, \dots, n\} \end{aligned}$$

where  $x_{ij}$  is the number of weapons of type  $i$  to assign to target  $j$ .

The Dynamic WTA (DWTA) is an extension of the SWTA and considers more than one engagement. While there are variants of the DWTA, such as the “shoot-look-shoot” model, this paper focuses on the multiple stage problem. In this approach, we have complete knowledge of a first stage which is defined by the SWTA. However, we know, to a probability distribution, the number and type of targets which will be



fired in a subsequent stage or stages. We define the formulation for the two stage problem, which can be extended to multiple stages, as

$$\begin{aligned}
Z_1 = \min_x & \left\{ \sum_{j=1}^{n_1} V_j^{(1)} \prod_{i=1}^m (1 - p_{ij})^{x_{ij}^{(1)}} + \mathbb{E}_{\omega \in \Omega} [Z_2(x^{(2)}, \omega^j)] \right\} \\
\text{st} & \sum_{j=1}^n x_{ij} \leq w_i \text{ for } i = 1, \dots, m, \\
& x_{ij} \in \mathbb{Z}_+, \text{ for } i = \{1, \dots, m\}, j = \{1, \dots, n\}
\end{aligned}$$

where  $n_1$  is the number of targets in the first stage with values  $V_j^{(1)}$  and  $x_{ij}^{(1)}$  is the number of weapons of type  $i$  to assign to target  $j$  in the first stage. We define the value of the second stage,  $Z_2$ , as a function of the remaining weapons,  $x^{(2)}$ , and a random occurrence,  $\omega \in \Omega$ , where  $\Omega$  is the set of all combinations of numbers and types of targets in the second stage.

$$Z_2(x^{(2)}, \omega^j) = \min_{x^{(2)}} \left\{ \sum_{j=1}^{n_2(\omega)} V_j^{(2)}(\omega) \prod_{i=1}^m (1 - p_{ij}(\omega))^{x_{ij}^{(2)}} \right\}.$$

The rest of this paper is presented as follows: in §2 we review some of the literature which we use as a basis from which to expand DWTA research. In §3, we define several solution techniques with which we solve the DWTA. In §4 we present two experiments in which we test the solution techniques from §3. We conclude this paper with a discussion of the results and their implications in §5.

## 6.2 Literature Review

The basic formulation for the SWTA, defined in §1, was the formulation defined by Manne (1958), who attributed it to Dantzig. Several solution techniques have found optimal solutions to this formulation. denBroeder et al. (1959) developed the

Maximum Marginal Return (MMR) algorithm, which is a greedy algorithm that finds the optimal solution when all weapons have the same probability of kill to target  $j$ . Ahuja et al. (2007) developed a branch and bound algorithm using a hybrid lower bounding strategy to solve a logarithmic transformation of the SWTA. Johansson & Falkman (2009) used a full enumeration algorithm to find exact solutions to small problem sizes. Kline (2017) developed a branch and bound algorithm to solve the untransformed SWTA.

Because the SWTA is NP-Complete (Lloyd & Witsenhausen, 1986), much more focus has been given towards efficiently finding near optimal solutions. There have been many implementations of the genetic algorithm, ant colony optimization algorithm, and MMR algorithm, and less frequently the very large scale neighborhood search heuristic, network flow heuristics, and tabu search heuristics. Kline et al. (2017) developed a heuristic with similarities to the Hungarian Algorithm which we modify and utilize in §3.

The DWTa receives less attention than the SWTA, and most of the work on the DWTa, like the SWTA, utilizes heuristic approaches to efficiently find near optimal solutions. One of the exact solution techniques is that of Ahner & Parson (2015), who use the Concave Adaptive Value Estimation (CAVE) Algorithm originally defined by Godfrey & Powell (2001) to converge to the optimal number of weapons to reserve for a second stage with the assumption that all weapons have the same probability of kill for target  $j$ .

### 6.3 Solution techniques

We develop a solution technique to the heterogeneous 2 stage DWTa which utilizes the CAVE Algorithm in a manner similar to Ahner & Parson (2015). We note that, while the solution given by Ahner & Parson (2015) is proven to be optimal, our

technique cannot guarantee optimality due to the differing probabilities of kill between weapon types, as illustrated by Kolitz (1988). As such, we utilize a subroutine, the Greedy Hungarian-like (GH) Heuristic, in our CAVE Algorithm which demonstrates superior performance to the MMR Plus Algorithm used by Ahner & Parson (2015) for the heterogeneous case. Further, as we recognize that the solution we find with our CAVE Algorithm is not guaranteed to converge to optimality, we build a Markov Decision Process (MDP), which we solve using backwards induction, that will converge to an optimal solution. We use the solutions of the MDP to identify the optimality gap of the solution of the CAVE Algorithm for small problem instances. However, if we had an optimal assignment algorithm in lieu of the GH heuristic, the CAVE Algorithm would converge to the optimal solution, as we prove in §3.1.

### **CAVE Algorithm.**

Our approach uses a subroutine for the CAVE Algorithm, the GH Heuristic, which assigns all weapons to available targets in the first stage while allocating weapons to a dummy target representing the approximated value of the second stage targets, which we derive via the CAVE Algorithm. In Figure 6, we define  $\mathbf{Y}^{(1)}$  as a matrix of the expected values of survival for all assignment pairings. We then define  $\mathbf{Y}^{(2)}$  as the difference between each element of  $\mathbf{Y}^{(1)}$  and the mean value of its row. Each element of  $\mathbf{Y}^{(3)}$  is the difference between its value in  $\mathbf{Y}^{(2)}$  and the minimum value of its column. Finally, the elements in  $\mathbf{Y}^{(4)}$  are the difference between their expected value of destruction and their value in  $\mathbf{Y}^{(3)}$ . We concatenate  $\mathbf{Y}^{(4)}$  with the values of the CAVE function with one additional weapon of each type,  $Z(x^{(2)} + \mathbf{1}, \omega)$ , and determine  $\{\hat{i}, \hat{j}\} = \arg \max[\mathbf{Y}^{(4)} \quad Z(x^{(2)} + \mathbf{1}, \omega)]$ . If  $\{\hat{i}, \hat{j}\}$  corresponds to a target in stage 1, we update  $x^{(1)}(\hat{i}, \hat{j}) = x^{(1)}(\hat{i}, \hat{j}) + 1$ . Otherwise, we update  $x^{(2)}(\hat{i}) = x^{(2)}(\hat{i}) + 1$  and repeat the process until all weapons have been assigned.

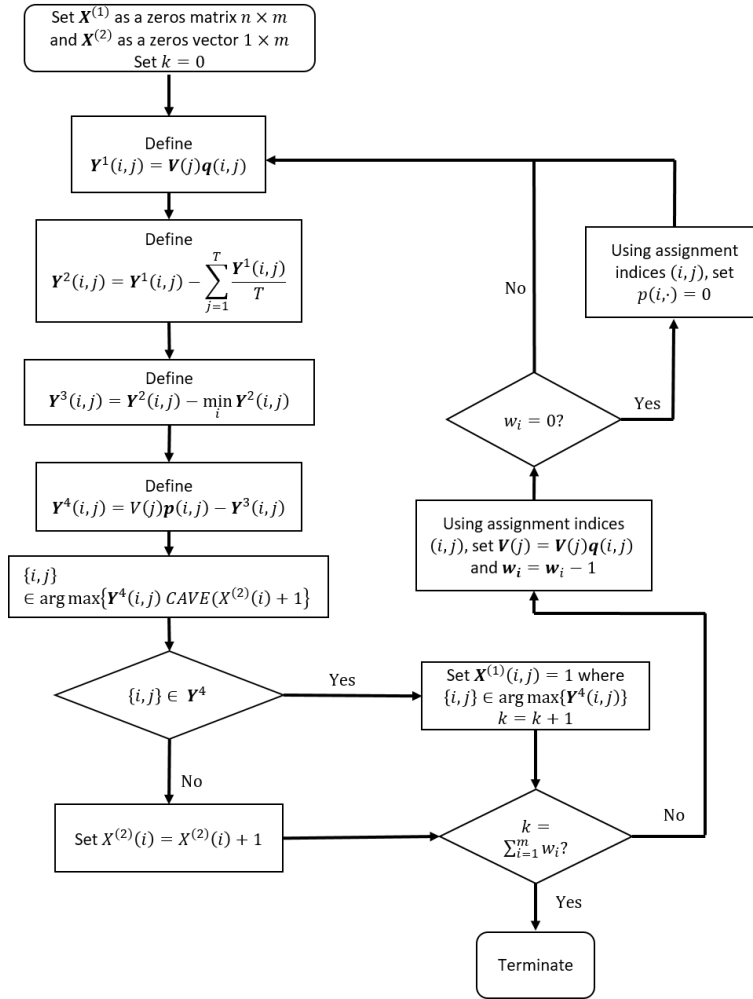


Figure 6. Greedy Hungarian-like Heuristic Flow Chart

The CAVE Algorithm generates a cost-to-go function, referred to as the CAVE function, which we use to determine how many of weapon type  $i$  to save for the second stage of the DWTa by means of our GH subroutine as previously described. We initialize the algorithm by first randomly generating parameters of our problem as uniformly distributed variables,  $p_{ij} \in U[p_{ij}^{low}, p_{ij}^{high}]$  and  $V_j \in U[V_j^{low}, V_j^{high}]$ , after which we execute the subroutine. We then generate a random sample of the number and type of targets in stage 2 and determines the subgradients

$$\begin{aligned}\nu^+(x^{(2)}(i), \omega) &= Z(x^{(2)}(i) + 1, \omega) - Z(x^{(2)}(i), \omega) \\ \nu^-(x^{(2)}(i), \omega) &= Z(x^{(2)}(i), \omega) - Z(x^{(2)}(i) - 1, \omega)\end{aligned}$$

where  $Z(\mathbf{X}^{(2)}(i), \omega)$  is the GH solution to the second stage for weapon type  $i$  given random instantiation  $\omega \in \Omega$ . Using these subgradients, we define a smoothing interval

$$I = [\max(0, \min(x^{(2)}(i) - \epsilon^-, u^{k^-})), \min(\max(x^{(2)}(i) + \epsilon^+, u^{k^++1}), M(i))]$$

where  $u^k$  is the breakpoint associated with  $x^{(2)}$  in the finite set of ordered breakpoints  $\{\nu^k, u^k | k \in K\}$ ,  $\epsilon$  is a step size governing how far out to smooth the CAVE function, and  $M(i)$  is the number of weapons available of type  $i$ . We perform smoothing over the interval  $I = [u^m, u^n]$  using a stepsize, which is a function of the iteration  $j$  we are performing in the CAVE Algorithm,  $\alpha = \frac{1}{1+j}$

$$\begin{aligned}\nu_{new}^k &= \alpha \nu^-(x^{(2)}(i), \omega) + (1 - \alpha) \nu_{old}^k & \text{for } k = m(i), \dots, x^{(2)}(i) - 1 \\ \nu_{new}^k &= \alpha \nu^+(x^{(2)}(i), \omega) + (1 - \alpha) \nu_{old}^k & \text{for } k = x^{(2)}(i), \dots, n(i) - 1\end{aligned}$$

We iterate the CAVE Algorithm until the either the change in subgradients across all weapon types is within some small tolerance or a predetermined number of itera-

tions have been performed. The end state of the CAVE Algorithm is the number of weapons of each type to preserve for the second stage.

Theorem 3.1 shows that, given an optimal assignment algorithm that can solve the problem in polynomial time, the algorithm would provide the optimal solution to the two stage problem with two weapon types. Unfortunately, no such algorithm is known for this NP-Hard problem.

**Theorem 3** *Assume the optimal solution for a two weapon type problem is  $(x_1^1, x_1^2, x_2^1, x_2^2)$  and that the optimal second stage subgradients which indicate preserving  $n_2^1$  and  $n_2^2$  weapons are bound by  $[D_1^-, D_1^+]$  and  $[D_2^-, D_2^+]$ . Given the assignment algorithm within the CAVE Algorithm is an optimal algorithm,  $w_1$  weapons of type 1, and  $w_2$  weapons of type 2, the CAVE Algorithm will generate the optimal solution  $(x_1^1, x_1^2, x_2^1, x_2^2)$ .*

PROOF. If we know the optimal number of weapons to preserve of either one of the weapon types, we can utilize Theorem 4.3 from Ahner & Parson (2015) to prove that the CAVE Algorithm will generate the optimal solution.

If we do not know the optimal number of weapons of either type to preserve a priori, the CAVE Algorithm will generate the optimal solution if  $\lambda_1 \in [D_1^-, D_1^+]$  and  $\lambda_2 \in [D_2^-, D_2^+]$  since the subroutine generates optimal assignments in each stage. Therefore, the CAVE Algorithm will not generate the optimal solutions if any of four cases occurs: when  $\lambda_1 < D_1^-$  and  $\lambda_2 < D_2^-$ , when  $\lambda_1 > D_1^+$  and  $\lambda_2 > D_2^+$ , when  $\lambda_1 < D_1^-$  and  $\lambda_2 > D_2^+$ , or when  $\lambda_1 > D_1^+$  and  $\lambda_2 < D_2^-$ :

Case 1:  $\lambda_1 < D_1^-$  and  $\lambda_2 < D_2^-$ . In this first case, the CAVE Algorithm finds less than  $w_1$  subgradients with values greater than or equal to  $D_1^-$  and less than  $w_2$  subgradients with values greater than or equal to  $D_2^-$ . At the optimal solution, there are  $w_1 - n_2^1$  subgradients with values greater than or equal to  $D_1^-$  and  $w_2 - n_2^2$  subgradients with values greater than or equal to  $D_2^-$ . Further, there are  $n_2^1 + 1$  and  $n_2^2 + 1$  subgradients in the second stage with values greater than or equal to  $\lambda_1$  and  $\lambda_2$ ,

respectively. However, since the CAVE Algorithm selects subgradients sequentially, it would require  $w_1 + 1$  and  $w_2 + 1$  weapons in order to converge at  $\lambda_1$  and  $\lambda_2$ .

Case 2:  $\lambda_1 > D_1^+$  and  $\lambda_2 > D_2^+$ . In this case, the CAVE Algorithm finds  $w_1$  and  $w_2$  subgradients with values greater than  $D_1^+$  and  $D_2^+$ . In the first stage at the optimal solution, there are  $w_1 - n_2^1$  and  $w_2 - n_2^2$  subgradients with values greater than or equal to  $D_1^+$  and  $D_2^+$ , respectively. In the second stage, there are  $n_2^1 - 1$  and  $n_2^2$  subgradients with values greater than or equal to  $D_1^+$  and  $D_2^+$ , respectively. This means that there are a total of  $w_1 - 1$  and  $w_2 - 1$  subgradients with values greater than or equal to  $D_1^+$  and  $D_2^+$ , respectively, which is a contradiction.

Case 3:  $\lambda_1 < D_1^-$  and  $\lambda_2 > D_2^+$ . We denote the subgradient proceeding  $\lambda_2$  as  $\lambda_2^-$ . Because we know that  $D_2^- \leq \lambda_2^- \leq D_2^+$  and  $\lambda_1 < D_1^-$ , it follows that  $\lambda_1 < \lambda_2^-$ . However, the CAVE Algorithm will not converge to subgradients  $(\lambda_1, \lambda_2)$  since it selects subgradients sequentially. Therefore, either  $\lambda_1 \geq D_1^-$  and  $\lambda_2 > D_2^+$  or  $\lambda_1 > D_1^-$  and  $\lambda_2 \leq D_2^+$ . Addressing the first of these cases, we showed in Case 2 that  $\lambda_1 > D_1^+$  and  $\lambda_2 > D_2^+$  is a contradiction and cannot occur. Additionally, if  $D_1^- \leq \lambda_1 \leq D_1^+$ , we showed that the CAVE Algorithm will find the optimal solution. Addressing the second of these cases, we showed in Case 1 that  $\lambda_1 < D_1^-$  and  $\lambda_2 < D_2^-$  is a contradiction and cannot occur. Further, if  $D_2^- \leq \lambda_2 \leq D_2^+$ , the CAVE Algorithm will find the optimal solution.

Case 4:  $\lambda_1 > D_1^+$  and  $\lambda_2 < D_2^-$ . This case is similar to Case 3 and can be proven in the same manner.

□

## MDP.

We model the heterogeneous 2 stage DWTA as an MDP in order to generate optimal solutions to smaller problem sizes. MDPs are characterized by the collection

of objects

$$\{\mathcal{T}, \mathcal{S}, \mathcal{A}_s, p_t(\cdot|s, a), r_t(s, a)\}$$

We define each of these objects here. The first,  $\mathcal{T}$ , denotes the planning horizon of the problem. We model the DWTA as a finite horizon problem, with decision epochs denoting which target to consider for interdiction. Given  $n$  targets in the first stage, we define the planning horizon to extend to  $n + 1$ , which is a dummy target which represents the value of the second stage.

$$\mathcal{T} = \{1, \dots, N + 1\}$$

The states of this model define the number of unassigned weapons of each type remaining. Given  $w_i$  weapons of type  $i$ , the state space is of cardinality  $(w_i + 1)^m$ , where  $m$  is the number of different weapon types.

The actions of this model define whether or not to assign any or all of the remaining weapons to a given target. An action of all zeros indicates that no weapons will be assigned to a target whereas one with all ones indicates that one of each weapon type is to be assigned. If any weapons remain, the actions available include all permutations of assigning or not assigning each of the available weapons, setting the size of the allowable actions to  $\prod_{i=1}^m (w_i + 1)$ , where  $w_i$  is the number of available weapons of type  $i$ . If we have a DWTA with 2 each of 5 weapon types, when all weapons are available there are  $3^5 = 243$  allowed actions. When one of one type of weapon has been assigned, there are  $3^4 * 2 = 162$  allowed actions.

The probability transition function deterministically defines the transition from the current state to the state reflecting the lack of the weapon(s) which was (were)



assigned in the previous stage.

$$p_t(j|s, a) = \begin{cases} 1 & \text{if } j = s - a \\ 0 & \text{otherwise} \end{cases}, \quad s \in S, a \in A_s, t = 1, 2, \dots, N - 1$$

So if we have one of each weapon type available,

$$S_t = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

and we assign one of weapon type 3,

$$A_s = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

we will transition to state

$$S_{t+1} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

with probability 1.

The reward functions for decision epochs  $\mathcal{T} = \{1, \dots, N\}$  are defined as the expected value of survival of a target given an assignment.

$$r_t(s, a) = V_t \prod_{i=1}^m q_{it}^{a_s(i)},$$

where  $q_{it}$  denotes the probability of weapon  $i$  missing target  $t$  and  $a_s(i)$  is the  $i^{th}$  element of action  $a_s \in A_s$ . However, the reward function for decision epoch  $N + 1$  is the optimal expected solution for each state for all possible second stage targets. That is, we must, for each state, determine the best assignment of available weapons for each possible configuration of the number and types of targets in the second stage.

Since we have an equal probability of each configuration, we take the average of the best assignment values for each state. This gives us the optimal expected value for each state in the  $N + 1$  decision epoch.

### **Baseline Policy.**

We develop a simple baseline policy which ensures that every target in the first stage has assigned to it one weapon. We reserve for the second stage all remaining weapons after allocating one to each target in the first stage. With the number of reserved weapons for the second stage known, we utilize the GH Heuristic to determine the objective function value of the first stage. Similarly, in computing the objective function value of the second stage, we utilize the GH Heuristic.

We utilize this rule-based policy as a baseline because it is accessible without the need for computationally complex algorithms or simulations. As we seek to demonstrate the solution quality that a dynamic programming based algorithm is capable of achieving, we compare it to a baseline policy which does not use dynamic programming.

## **6.4 Computational Results**

We test the aforementioned solution techniques across two experimental designs. The smaller of the two is extensive enough to illustrate the differences in performance between the CAVE Algorithm and the baseline policy but is also small enough that the MDP can converge to the optimal solution without exceeding available memory, which is the result of defining the action space in computing the terminal reward. Solving the MDP allows for the comparison of policies by their optimality gaps. The other experimental design is larger and finding an optimal policy to any of these instances is intractable.

Within each of the two experimental designs, we define the target value and the weapon-target probability of kill parameters as follows. For each target, a uniformly distributed continuous variable value  $V_j \in [25, 100]$  is randomly generated, and we also assigned randomly generated probabilities of kill as uniformly distributed continuous variables  $p_{ij} \in [0.6, 0.9]$  so that each weapon has a different probability of kill for each target.

The parameters that we vary among the two experimental designs are as follows. We vary the numbers of weapon types,  $m$ , each type with a different number of weapons,  $w_i$ . We also vary the number of target types, which is also the number of targets in the first stage,  $n_1$ . Lastly, we vary the number of possible targets in the second stage. We do this by defining the minimum number of targets in the second stage,  $t_2^{\min}$ , and the range of the number of targets in the second stage,  $t_2^{\text{range}}$ . So an instance with the following parameters

$$\begin{bmatrix} m \\ w_1 \\ w_2 \\ n_1 \\ t_2^{\min} \\ t_2^{\text{range}} \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 3 \\ 5 \\ 2 \\ 3 \end{bmatrix},$$

we would have 2 different weapon types, the first with 4 weapons and the second with 3 weapons. Further, we would have 5 targets in the first stage and anywhere from 2 through 5 targets in the second stage, which occur according to a triangle distribution. The targets in the second stage are selected randomly from the target types, which means that the target values and the associated weapon-target probabilities of kill correspond to those for each target type in the first stage.

We generate 30 problem instances using random numbers for each design point and performed all tests on a computer having an Intel Xeon E5-2650 v2 processor with 128 GB RAM. Each solution method is applied to the same set of 30 problem to allow for direct comparisons of solution values and computational times. Our results, insights, and analysis are presented in this section.

### **Small Experimental Design.**

We define an experiment wherein we can find optimal policies as determined by the MDP model described in §3.2 as shown in Table 22. For these smaller design points, we compute the optimal policy using the MDP algorithm and offer a comparison to the CAVE Algorithm and baseline policy solutions by determining the first stage solution with the given policies and computing the solution for every permutation of the second stage possible. That is, we determine the objective function value of each realization of each permutation of target types in each possible number of targets in the second stage and compare the average performance of each solution technique to the optimal policy of the MDP algorithm.

As the convolution of random numbers leaves us unable to assume a normal distribution among parameters, we implement a nonparametric statistical test, the Kruskal-Wallis test, to assess the performance of the solution methods. We test two null hypotheses: (1) the CAVE Algorithm is equivalent to the baseline policy, and (2) the optimal MDP policy is equivalent to the CAVE Algorithm policy. We can see in Table 22 that, with a confidence of  $\alpha = 0.05$ , 9 of the 11 problem instances reject null hypothesis 1. In 8 of these 9 instances, the CAVE Algorithm outperforms the baseline policy. We also observe that 9 of the 11 instances reject null hypothesis 2. In instance 8, which does not fail to reject null hypothesis 2, the optimality gaps of the 30 runs for the CAVE Algorithm do not deviate far from 99%, thus we reject the

**Table 22. Small Experimental Design**

Instance	Parameter Settings							Experimental Results			
	$m$	$n_1$	$t_2^{\min}$	$t_2^{\text{range}}$	$w_1$	$w_2$	$w_3$	Algorithm	Avg Opt Gap (%)	Avg Imp over Base (%)	Comp Time (sec)
1	2	5	3	1	2	3	-	MDP	N/A	35.51	244.86
								CAVE	84.17	14.11	6.20
								Baseline	73.90	N/A	0.0011
$2^\dagger$	2	4	3	2	2	2	-	MDP	N/A	20.67	238.72
								CAVE	99.59	20.18	6.35
								Baseline	83.20	N/A	0.00049
3	2	4	1	1	3	3	-	MDP	N/A	0.35	0.12
								CAVE	95.80	-3.87	5.15
								Baseline	99.66	N/A	0.00035
4	2	4	2	2	3	2	-	MDP	N/A	20.34	92.69
								CAVE	87.65	5.48	5.62
								Baseline	83.12	N/A	0.00049
5	3	5	2	1	2	2	3	MDP	N/A	14.82	390.21
								CAVE	88.05	1.10	11.15
								Baseline	87.11	N/A	0.0014
6	3	3	1	2	3	2	2	MDP	N/A	10.18	69.44
								CAVE	93.31	2.80	9.30
								Baseline	90.78	N/A	0.00090
$7^*$	3	5	1	1	3	3	2	MDP	N/A	5.77	2.56
								CAVE	94.97	0.44	6.20
								Baseline	94.58	N/A	0.00043
$8^*$	2	3	2	2	3	3	-	MDP	N/A	0.42	30.00
								CAVE	99.65	0.064	8.73
								Baseline	99.59	N/A	0.00041
9	2	4	2	1	3	2	-	MDP	N/A	18.98	2.70
								CAVE	87.83	4.50	7.45
								Baseline	84.07	N/A	0.00032
10	2	5	1	2	2	3	-	MDP	N/A	22.24	5.53
								CAVE	89.86	9.86	6.97
								Baseline	81.82	N/A	0.00042
$11^\dagger$	2	3	2	1	2	2	-	MDP	N/A	6.94	0.26
								CAVE	98.51	5.38	5.79
								Baseline	93.61	N/A	0.00027

\* fail to reject null hypothesis 1

$^\dagger$  fail to reject null hypothesis 2

null hypothesis despite its close proximity to the MDP policies.

We note that the average computational time for the three algorithms do not consistently favor the CAVE Algorithm over the MDP algorithm. This is because the CAVE Algorithm observes many realizations of the second stage in order to approximate its value function, which may be more computationally expensive than an MDP with a relatively small action space. By contrast, we see that the computational time required for some of the instances far exceeded the CAVE Algorithm. Further, there were experimental settings for which the MDP did not converge to an optimal policy within four week’s continuous computation, whereas the CAVE Algorithm was able to find a solution within 20 seconds.

**Table 23. Solutions within n% Optimal (%)**

	$\leq 70$	$[70, 80)$	$[80, 90)$	$[90, 100)$	Optimal
CAVE	0	0.30	35.15	56.06	8.48
Baseline	0.61	11.82	44.55	37.58	5.45

We see in Table 23 the percentage of solutions for the CAVE Algorithm and the Baseline policy by their optimality. Roughly 57% of the solutions found by the Baseline policy, which requires minimal computational effort, are less than 90% optimal whereas roughly 35.5% of the solutions found by the CAVE algorithm are less than 90% optimal (only 1 solution was less than 80% optimal). By contrast, roughly 43% of the solutions found by the Baseline policy are 90% optimal or better as compared to the roughly 64.5% of solutions found by the CAVE algorithm which are 90% optimal or better. This allows us to conjecture two points about the solution techniques. First, the CAVE Algorithm generally finds better solutions than the Baseline policy, as indicated by Table 22 and Table 23. Second, the CAVE Algorithm is a more reliable solution technique than the Baseline policy, whose solutions vary between 68% optimal and 100% optimal.

## Large Experimental Design.

We define an experiment wherein we cannot find optimal policies as determined by the MDP model described in §3.2, but rather test the performance of the CAVE Algorithm as compared to the baseline policy for larger design points, as shown in Table 24. For these larger design points, we compute the solution to the first stage and randomly sample 10,000 realizations of the second stage, computing the objective function value of each realization with the CAVE Algorithm policy and baseline policy. We use these solutions to determine the expected value of each policy.

As in §4.1, we use the Kruskal-Wallis test to assess the performance of the CAVE Algorithm as compared to the baseline policy. We assume that the CAVE Algorithm has similar optimality gaps for the larger problem instances to those of the smaller problem instances, wherein it found policies which are 92.7% optimal. We see in Table 24 that, with a confidence of  $\alpha = 0.05$ , we can reject null hypothesis 1 in 28 of the 33 instances, and that for each of these 28 instances, the average improvement of the CAVE Algorithm over the baseline policy is positive, therefore demonstrating statistically superior solutions in the CAVE Algorithm in these instances.

## 6.5 Conclusion

The two stage heterogeneous WTA is a complex problem which has received limited in the literature. Each stage is an NP-Hard problem, with the number of weapons in the second stage dependent upon the assignment made in the first stage. An optimal policy, which is achievable for small problems, is computationally expensive. Indeed, a single problem with 3 weapon types (2 weapons of type 1 and 3 weapons of type 2 and 3), 4 targets in the first stage, and 2-4 targets in the second stage did not converge to an optimal solution after four weeks of computation using an MDP backwards induction algorithm. The use of a computationally simple policy, which we call

**Table 24. Large Experimental Design**

Instance	Parameter Settings									Experimental Results	
	$m$	$n_1$	$t_2^{\min}$	$t_2^{\text{range}}$	$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	Avg Imp over Base (%)	CAVE Time (sec)
1	5	11	10	4	9	8	8	7	10	4.82	37.77
2	5	20	5	6	7	6	9	7	10	3.91	28.18
3	5	14	18	4	5	8	9	5	7	0.049	101.27
4	4	19	20	6	10	6	9	5	-	4.95	67.35
5	5	10	11	5	8	9	7	8	5	2.36	37.68
6	5	19	8	5	7	6	6	9	5	0.11	45.33
7	4	15	19	5	5	8	7	10	-	1.63	56.92
8	4	17	19	5	10	6	6	10	-	1.33	66.96
9	4	13	7	7	9	7	5	6	-	0.31	30.88
10	4	17	8	8	6	8	5	7	-	0.46	45.62
11	4	12	16	10	7	5	6	6	-	2.14	66.41
12*	4	17	14	10	9	10	7	7	-	0.0020	61.06
13	4	12	6	7	8	6	10	9	-	4.07	23.10
14	5	16	9	9	6	8	10	9	6	1.88	40.42
15	4	12	17	9	7	5	8	9	-	0.40	64.71
16*	4	16	13	10	9	10	8	8	-	0.030	56.23
17*	4	15	12	7	8	8	8	8	-	-0.046	42.57
18	2	19	13	9	6	7	-	-	-	14.66	13.30
19	2	10	18	7	8	9	-	-	-	7.75	22.35
20	2	16	5	9	10	7	-	-	-	12.50	11.46
21	3	11	3	7	5	9	6	-	-	0.82	16.49
22	2	20	12	8	7	6	-	-	-	12.35	10.39
23	2	11	15	8	8	9	-	-	-	8.35	22.41
24	3	15	4	8	10	7	8	-	-	0.25	16.51
25	3	13	4	8	5	9	9	-	-	0.79	18.26
26	3	18	16	6	6	8	10	-	-	9.18	33.83
27	3	13	15	5	9	7	10	-	-	1.05	32.97
28*	3	18	7	3	8	10	9	-	-	-0.15	23.29
29	3	13	9	3	6	5	8	-	-	3.05	25.50
30	3	18	17	6	7	9	5	-	-	17.14	35.97
31	2	14	14	4	9	7	-	-	-	18.54	16.89
32*	3	18	6	4	8	10	7	-	-	-0.021	21.72
33	3	14	10	3	6	5	7	-	-	8.05	24.41

\* fail to reject null hypothesis 1



a baseline policy, converges to a solution rapidly but is unreliable in its performance, with solutions as low as 68.6% optimal. As such, we develop a computationally simple technique by which a near optimal solution is reliably found.

We present a solution technique for the heterogeneous DWTA which uses the CAVE Algorithm with the Greedy Hungarian-like Heuristic as a subroutine. We compare the performance of this algorithm to the performance of a baseline policy and, when possible, to the performance of an optimal MDP algorithm. Though limited by the size of the problem instance considered, this MDP algorithm is the first optimal solution technique for the heterogeneous DWTA found in the literature.

We extend the work of Ahner & Parson (2015), who use the CAVE Algorithm with an optimal subroutine to find the optimal solution to the homogeneous DWTA. Our work is able to find near optimal policies to a two stage problem with different types of weapon systems for small problem instances. The number of dimensions to consider in each stage grows exponentially with the number of weapon systems, which increases the required computational time for both the CAVE and the MDP algorithms. The CAVE Algorithm is able to generate a policy which improves upon a baseline policy within two minutes for the largest problem instances considered and is easily extended to large problem instances.

## VII. A Continuous Time Two Stage Shoot-Look-Shoot Weapon Target Assignment Problem

### 7.1 Introduction

With many variants and over sixty years (Manne, 1958) of consistent research, the defensive weapon target assignment (WTA) problem is one whose solution informs the targeting of incoming missiles with available interceptors in defense of a protected assets. In his seminal work, Manne modeled what is now referred to as the static WTA (SWTA), which provides a scenario wherein a known number of incoming missiles (targets) with known destructive values is to be intercepted by a number of interceptors (weapons) with known probabilities of successfully destroying the missiles (probabilities of kill). With  $j = 1, \dots, n$  targets, each of which has a destructive value  $V_j$ , and  $i = 1, \dots, m$  weapons, with probabilities of kill  $p_{ij}$ , the formulation is

$$\begin{aligned} \min \quad & \sum_{j=1}^n V_j \prod_{i=1}^m (1 - p_{ij})^{x_{ij}} \\ \text{st} \quad & \sum_{j=1}^n x_{ij} \leq w_i, \text{ for } i = 1, \dots, m, \\ & x_{ij} \in \mathbb{Z}_+, \text{ for } i = 1, \dots, m, \ j = 1, \dots, n \end{aligned} \tag{1}$$

where the decision variable,  $x_{ij}$ , defines the number of weapons of type  $i$  to assign to target  $j$ . The first constraint limits the number of weapons of type  $i$  that can be assigned to the total number of weapons of type  $i$  available,  $w_i$ . The second constraint limits the decision variable to positive integers.

Advances in WTA modeling and computational power enabled the development of a dynamic WTA (DWTA), in which engagements can either be observed and repeated (if the weapon misses the target) in a “Shoot-Look-Shoot” model, or a subsequent stage in which a number of targets known only to a probability distribution can be

fired at the protected asset in a “Two Stage” model. The homogeneous Two Stage DWTA, which was initially proposed by Murphey (2000) and later optimally solved by Ahner & Parson (2015), seeks to minimize the sum of the expected values of survival for the targets in the first stage and the expected number of targets in the second stage, which occurs according to a random variable  $\omega \in \Omega$ , where  $\Omega$  is the set of all possible outcomes of remaining targets whose distribution has a finite second moment.

$$\begin{aligned} \min \quad & \left\{ \sum_{j=1}^{n_1} V_j^{(1)} \left(1 - p_{ij}^{(1)}\right)^{x_{ij}^{(1)}} + \mathbb{E}_{\omega \in \Omega} \left[ Z_2 \left( x^{(2)}, \omega^j \right) \right] \right\} \\ \text{st} \quad & \sum_{j=1}^n x_{ij} \leq w_i, \text{ for } i = 1, \dots, m, \\ & x_{ij} \in \mathbb{Z}_+, \text{ for } i = 1, \dots, m, \quad j = 1, \dots, n \end{aligned} \tag{2}$$

where the second stage,  $Z_2$ , is a function of the remaining weapons,  $x^{(2)}$ , and a random occurrence,  $\omega$ , of the number and type of targets.

$$Z_2 \left( x^{(2)}, \omega^j \right) = \min \left\{ \sum_{j=1}^{n_2(\omega)} V_j^{(2)}(\omega) \left(1 - p_{ij}^{(2)}\right)^{x_{ij}^{(2)}} \right\}.$$

While there are many exact and heuristic solution techniques for the SWTA and DWTA, there is relatively little research dedicated to the scheduling of these solutions. As no interceptor system can instantly engage multiple targets simultaneously, it is necessary to provide a firing order that governs the sequence in which the solution is to be executed. Khosla (2001) proposed a model which addressed this by assigning weapons to targets at specific time points in order to establish a solution which provides a feasible firing order. However, as he notes, the length of time to consider and the size of the time intervals to utilize can quickly increase the dimensionality of the problem to a point at which finding a solution is intractable.

An alternative model whose solution provides a feasible firing order was developed by Leboucher et al. (2013). By using two dimensional Bézier curves, Leboucher et al. (2013) are able to model flight paths of targets which define their position in continuous time. The model is a homogeneous variant, which means that all available weapons are of the same type. This contrasts with a heterogeneous variant, in which different weapon types, with different probabilities of kill, are available. Leboucher et al. (2013) implement a two step technique to solve their model, using the Hungarian Algorithm to determine the assignment solution and a hybrid particle swarm optimization and evolutionary game theory heuristic to determine the firing order.

In this paper, we present a heterogeneous model, based on the aforementioned model developed by Leboucher et al. (2013), which we extend to two stages while at the same time allowing for a Shoot-Look-Shoot approach and modeling continuous flight paths with three dimensional Bézier curves. In section §2, we present our model and identify our assumptions. In §3, we describe our solution technique, which involves several heuristics and an adaptation of the Concave Adaptive Value Estimation (CAVE) Algorithm developed by Godfrey & Powell (2001). We test our solution technique in §4 by running real-time simulations which initiate at the launch of the targets and requires we find a solution and conduct all engagements prior to the point of impact of each target over the number of stages. Finally, we analyze the results and provide our conclusions in §5.

## 7.2 The Model

We present a formulation which is an amalgamation of the two stage model developed by Ahner & Parson (2015) and the work of Leboucher et al. (2013). We define

our sets, in accordance with those described in §1, as follows:

$$I = \{1, \dots, m\} \quad \text{Weapon types}$$

$$J = \{1, \dots, n\} \quad \text{Target types}$$

$$K = \{1, \dots, a\} \quad \text{Protected assets}$$

$$T = \{1, \dots, b\} \quad \text{Stage.}$$

We define a fifth set, which is composed of the set of assignments in the solution for stage  $t$ , as

$$X^t = \{(i_1, j_1), \dots, (i_m, j_n)\}$$

Next, we define the parameters of our model. First, we define the probability that target  $j$  will hit and destroy protected asset  $k$  as  $\gamma_{jk}$ , or probability of hit. We define the probability that weapon  $i$  will hit and destroy target  $j$  as  $p_{ij}$ , or probability of kill. We define the time required to fire, guide, and prepare a weapon system for a subsequent engagement as our guide time,  $g$ . We define the number of weapons of type  $i$  available as  $w_i$ . Weapon  $i$  is located at  $l_i$  and it has an effective range of  $r_i$ . Target  $j$  is launched from a point of origin at a distance no closer than  $d_{jk}$  to protected asset  $k$  and can climb to a maximum altitude of  $h_j$ . In stage  $t$ , target  $j$  will be within the maximum effective range of weapon type  $i$  at time  $c_{ijt}^r$  and will have a time until impact at protected asset  $k$  of  $c_{ijkt}^s$ .

For the two stage model, we have additional parameters. We define the minimum number of targets in the second stage to be  $n_2^{min}$ , and we define the maximum number of targets in the second stage to be the sum of the minimum and  $n_2^r$ . So the number of targets in the second stage is randomly selected within the range  $n_2^{min} \leq n_2 \leq n_2^{min} +$

$n_2^r$ . The complete list of parameters for this model is as follows:

$\gamma_{jk}$	Probability of hit
$p_{ij}$	Probability of kill
$g$	Guide time
$M_k$	Value of protected asset $k$
$w_i$	Number of weapons of type $i$
$l_i$	Location of $i$
$r_i$	Range of $i$
$d_{jk}$	Max distance between $j$ and $k$
$h_j$	Max altitude of $j$
$c_{ijt}^r$	Time until in range
$c_{ijkt}^s$	Time until strike
$n_2^{min}$	Min number of targets in stage 2
$n_2^r$	Range of number of targets in stage 2

We define our decision variables  $x_{ij}^t$  and  $y_{ijq_i}^t$  as follows. The first decision variable,  $x_{ij}^t$ , indicates the number of weapons of type  $i$  assigned to target  $j$  in stage  $t$ . In the one stage problem, this can be simplified to  $x_{ij}$  since  $t = 1$  for all available assignments. The second decision variable,  $y_{ijq_i}^t$ , indicates the time at which a weapon of type  $i$  is assigned to target  $j$  in stage  $t$ . If  $x_{ij}^t > 1$ , we identify successive shots with the index  $q_i = 1, \dots, w_i$ . That is, if we have a non-zero value for  $y_{ij3}^t$ , then at least 3 weapons of type  $i$  are assigned to target  $j$  in stage  $t$ , with firing times  $y_{ij1}^t$ ,  $y_{ij2}^t$ , and  $y_{ij3}^t$ .

Using the aforementioned sets, parameters, and decision variables, we can define our model. We are seeking to minimize the expected value of any leakers, or targets which survive our defensive efforts. We assume that no target can destroy more than one protected asset, thus we define a target value  $V_j = \max \{M_k \gamma_{jk}\}$ . Therefore, we

have

$$\begin{aligned}
\min \quad & \left\{ \sum_{j=1}^{n_1} V_j^{(1)} \left( 1 - p_{ij}^{(1)} \right)^{x_{ij}^{(1)}} + \mathbb{E}_{\omega \in \Omega} \left[ Z_2 \left( x^{(2)}, \omega^j \right) \right] \right\} & (3) \\
\text{st} \quad & \sum_{t=1}^b \sum_{j=1}^n x_{ij}^t \leq w_i, & \text{for } i \in I, j \in J, t \in T, & (3a) \\
& \left| y_{ijq_i}^t - y_{ijq'_i}^t \right| \geq g, & \text{for } (i, j) \in X^t, t \in T, q_i, q'_i = 1, \dots, w_i, q'_i \neq q_i, & (3b) \\
& \left| y_{ijq_i}^t - y_{i'j'q'_i}^t \right| \geq g, & \text{for } (i, j), (i', j') \in X^t, t \in T, q_i, q'_i = 1, \dots, w_i, j' \neq j, & (3c) \\
& y_{ijq_i}^t \geq c_{ijt}^r, & \text{for } (i, j) \in X^t, t \in T, q_i = 1, \dots, w_i, & (3d) \\
& y_{ijq_i}^t \leq c_{ijt}^r + c_{ijkt}^s, & \text{for } (i, j) \in X^t, t \in T, q_i = 1, \dots, w_i, & (3e) \\
& x_{ij}^t \in \mathbb{Z}_+, & \text{for } i \in I, j \in J, t \in T, & (3f) \\
& y_{ijq_i}^t \geq 0, & \text{for } i \in I, j \in J, t \in T, q_i = 1, \dots, w_i, & (3g)
\end{aligned}$$

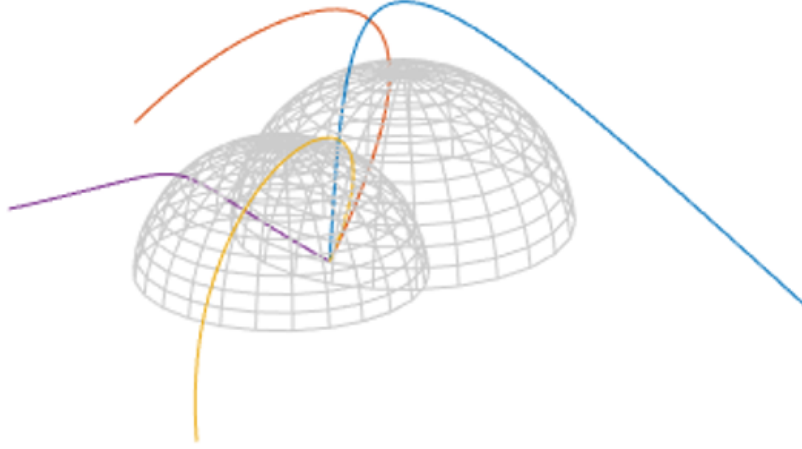
where the second stage,  $Z_2$ , is a function of the remaining weapons,  $x^{(2)}$ , as described in §1. In the above formulation, the objective function, (3), is equivalent to (2) from §1. Constraint (3a) limits the total number of assigned weapons of type  $i$  over all stages to  $w_i$ . Constraints (3b) and (3c) ensure that two engagements occur at least  $g$  time units apart unless fired from two weapon types at the same target. Constraint (3d) limits the earliest fire time of a weapon of type  $i$  to target  $j$  to the time until target  $j$  is within range of weapon  $i$ . Constraint (3e) ensures that the latest firing time of a weapon of type  $i$  to target  $j$  is the time of impact of target  $j$  at protected asset  $k$ . Lastly, constraints (3f) and (3g) limit the decision variables  $x_{ij}^t$  and  $y_{ijq_i}^t$  to positive integers and nonzero reals, respectively

We utilize Bézier curves to generate realistic flight paths of targets, as referenced in §1. In order to capture the three dimensional reality of a weapon's range and the flight path of a target, we extend the two dimensional Bézier curves to three dimensions. For each target, we generate three random control points and use the location of the targeted protected asset as the fourth control point. With control point  $n$  defined by  $P_n \equiv (x_n, y_n, z_n)$ , we generate each curve by using  $i = 1, \dots, m$

break points, computing the coordinates at each break point as

$$f(x_i, y_i, z_i) = \left(1 - \frac{i}{m}\right)^3 P_1 + 3 \left(1 - \frac{i}{m}\right)^2 \left(\frac{i}{m}\right) P_2 + 3 \left(1 - \frac{i}{m}\right) \left(\frac{i}{m}\right)^2 P_3 + \left(\frac{i}{m}\right)^3 P_4$$

Using the Bézier curves and the parameters  $l_i$  and  $r_i$ , we generate problem instances. Shown in Figure 7 is such a problem instance with two weapon types and four targets aimed at a single protected asset. The gray hemispheres represent the effective range of each weapon whereas the four curved lines indicate the flight paths of the four targets, which follow the paths with constant speed.



**Figure 7. 3-D Model using Bézier Curves**

As shown above, this model captures the continuous time aspect of the target flight paths and incorporates a multi-stage objective. We make several assumptions in developing this model. First, we assume that, once launched, each target has a constant speed and that all target velocities are equal. Next, we assume that we have full knowledge of the flight paths of the targets and can accurately assess the geometry of each target's descent to the point of impact at the protected asset, which we use to compute probabilities of hit and of kill. Next, we assume that a target of type  $j$  in the second stage will have the same flight path and associated parameters



as a target of type  $j$  in the first stage.

### 7.3 The Solution Techniques

We develop a solution technique, which we call the Continuous Reallocation Method (CRAM), capable of finding real-time solutions in the combined Shoot-Look-Shoot and two-stage model. Similar to Leboucher et al. (2013), we use a two step technique for each stage, solving first the assignment problem and subsequently determining the firing order. However, whereas Leboucher et al. (2013) determine the assignments and firing order once, the CRAM determines the assignments and firing order after every engagement. Because this element of control theory underlies the CRAM, it is an example of model predictive control (Ahner, 2005). For the two stage problem, we use a Concave Adaptive Value Estimation (CAVE) algorithm to determine the number of interceptors to use in the first stage and the number to save for the second stage. Herein, we present the heuristic algorithms that comprise the CRAM, through which we are able to solve the problem.

#### A Single Stage.

We first define the process by which we solve the model in a single stage as the two stage problem is an extension of the single stage problem. Given the parameters defined in §2, we first assign the available weapons to targets by solving the formulation in Equation (1) using a Greedy Hungarian-Like Algorithm (Kline et al., 2017). Next, we assign a firing order by first identifying the time at which each target will be within range of its assigned weapon. We then identify any assignments which are within range at the current clock time and denote this time as the firing time for the assignment within range which has the smallest time to impact. We increment the clock by a set guide time and repeat the process until the clock time exceeds the

latest time to impact. If no assignments are within range at the current clock time, we increment the clock by the guide time and repeat the process.

With the assignment and firing order algorithms described above, we solve the one stage model according to the CRAM as shown in Figure 8. Once the parameters are generated, the clock is started and the assignments and firing order are determined. Next, the procedure simulates the first weapon-target assignment according to the firing order, generating a random number,  $\omega \in [0, 1]$  to assess whether weapon  $i$  destroys target  $j$ . If  $\omega > p_{ij}$ , target  $j$  survives the engagement. Otherwise, target  $j$  is destroyed by weapon  $i$ . The clock is incremented by the guide time and the assignment and firing order heuristics update the engagement plan given knowledge of the disposition of the target. If it has been destroyed, the target no longer receives any weapons and the successful engagement is annotated. Otherwise, the assignments and firing order will adjust to account for the reduction of available weapons and time. This process continues, with the clock tracking the actual time of the simulation, until the last target is destroyed or has reached its destination. A different random number  $\omega \in [0, 1]$  for each surviving target determines whether it hit the protected asset. If  $\omega > \gamma_{jk}$ , target  $j$  misses protected asset  $k$ . Otherwise, target  $j$  hits protected asset  $k$ .

### **Two Stages.**

We extend the single stage CRAM to solve a two stage problem. In order to do this, we first must determine how many weapons to use in the first stage and how many to save for the second stage. We extend and implement the CAVE algorithm, used by Ahner & Parson (2015) to the two stage heterogeneous DWTA, as was first demonstrated by Kline et al. (2018).

The two stage CRAM begins by defining the parameters and Bézier curves, starting the clock, and using the CAVE algorithm to determine the number of weapons

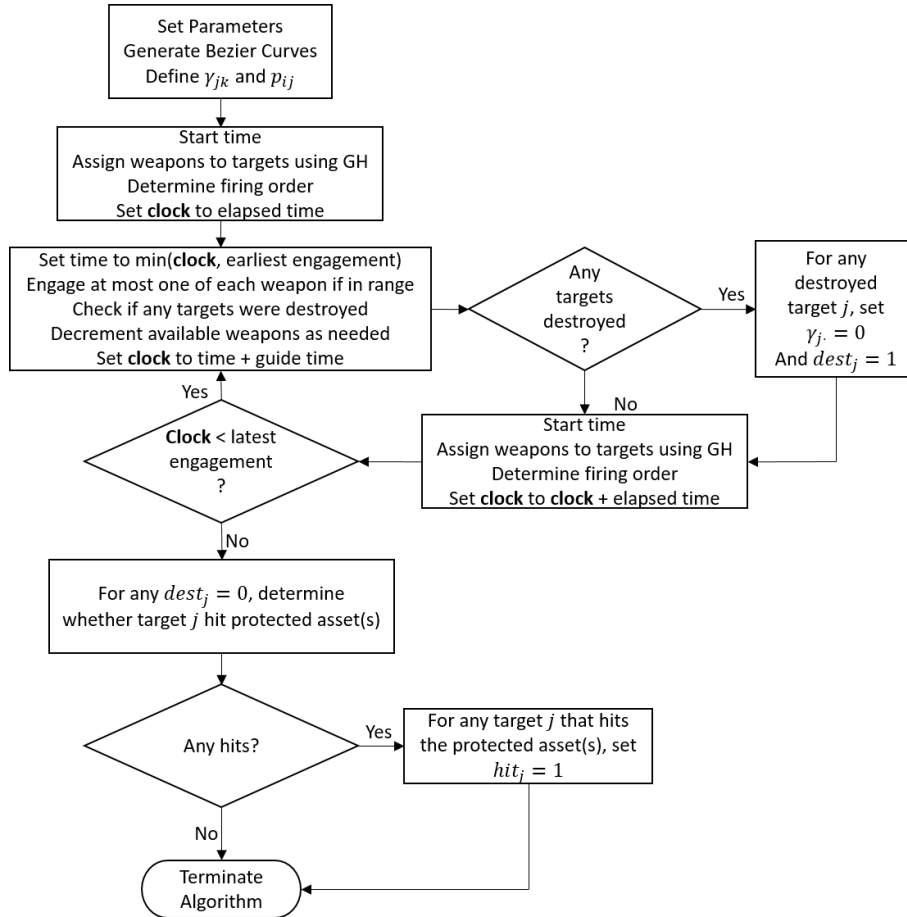


Figure 8. One Stage Procedure

to use in each stage. Next, the first stage is solved using the procedure defined in §3.1. Upon completion of the first stage, the number and type of targets in the second stage is randomly determined and the associated parameters are generated. The same procedure is then used for the second stage. The number of targets destroyed and the number of targets that hit the protected asset are determined just as in the one stage problem.

### **Baseline Policy.**

In order to assess the efficacy of our solution techniques for this problem, we develop a realistic Shoot-Shoot-Look policy which we call our Baseline Policy. In this policy, a target  $j$  is engaged with two of weapon  $i$  upon entering its effective range. If target  $j$  enters the effective range of weapon  $i$  and  $w_i = 1$ , the second engagement occurs with the next closest weapon once within range. If  $w_i = 0$ , the target is engaged with two weapons of the next closest weapon once within range. After each engagement, the survival of the target is assessed and the target is reengaged if it hasn't yet reached the protected asset. For the two stage problem, half of the weapons are saved for the second stage, with the extra weapon going to the second stage where there are an odd number of weapons of any type.

## **7.4 Computational Results**

We conduct two experiments to assess the performance of the solution techniques outlined in §3. The first experiment tests the performance in a single stage problem while the second tests the performance in a two stage problem. We use a nearly orthogonal Latin hypercube (NOLH) design for each experiment, varying the upper and lower bounds for both  $\gamma_{jk}$  and  $p_{ij}$ , guide time  $g$ , and number of weapons of type  $i$  available  $w_i$ . For the two stage problem, we also vary  $t_2^{min}$  and  $t_2^r$ . Within each

experiment, we do not vary the ranges ( $r_i$ ) of each of the weapon types we use, nor the max altitude ( $h_j$ ) or max distance ( $d_{jk}$ ) of target  $j$ , of which there are 16 in the first stage. Target  $j$  can have a launch point as far as 100 units in either direction, thus the maximum distance for a launch point is roughly 141.4 units away ( $\sqrt{100^2 + 100^2}$ ). All targets travel at a constant speed of 1 unit per second, so the longest possible simulation is 141.4 seconds, or roughly 2.3 minutes.

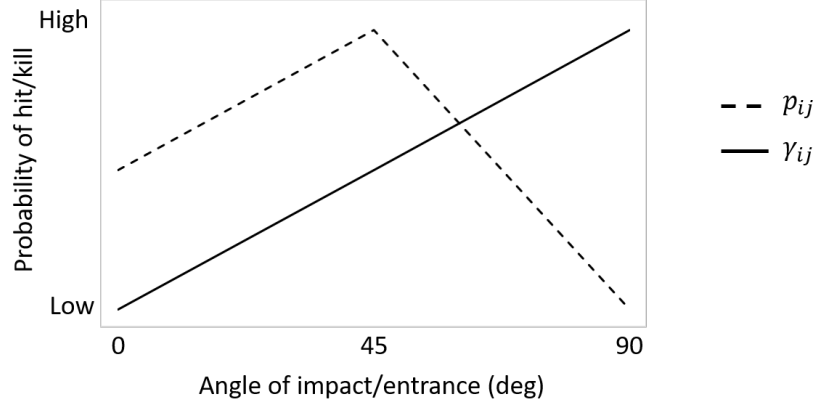
For each experiment, we build our NOLH design by bounding the parameters as follows

$$\begin{aligned}
\gamma_{jk} &\geq \gamma_{jk}^{low} & 0.15 &\leq \gamma_{jk}^{low} \leq 0.3 \\
\gamma_{jk} &\leq \gamma_{jk}^{high} & 0.35 &\leq \gamma_{jk}^{high} \leq 0.75 \\
p_{ij} &\geq p_{ij}^{low} & 0.25 &\leq p_{ij}^{low} \leq 0.50 \\
p_{ij} &\leq p_{ij}^{high} & 0.50 &\leq p_{ij}^{high} \leq 0.90 \\
g & & 0.5 &\leq g \leq 3 \\
w_i & & 5 &\leq w_i \leq 15.
\end{aligned}$$

We generate the values for  $\gamma_{jk}$  and  $p_{ij}$  by using the angle of impact or the angle by which the target enters the effective range of a weapon, respectively. We make a general assumption that the higher the angle of impact, the greater the probability of hit, whereas the closer to  $45^\circ$ , the higher the probability of kill, with a much lower probability of kill at  $90^\circ$  than at  $0^\circ$ , as can be seen in Figure 9. For a target  $j$  with an angle of impact/entrance  $\theta$  and a random deviation  $\epsilon$ , the parameters are:

$$\begin{aligned}
\gamma_{jk} &= \gamma_{jk}^{low} + \frac{\theta}{90^\circ} \left( \gamma_{jk}^{high} - \gamma_{jk}^{low} \right) + \epsilon \\
p_{ij} &= \begin{cases} \frac{(p_{ij}^{high} + p_{ij}^{low})}{2} + \frac{\theta}{90^\circ} \left( p_{ij}^{high} - p_{ij}^{low} \right) + \epsilon & \theta \leq 45^\circ \\ p_{ij}^{high} - \frac{\theta - 45^\circ}{45^\circ} \left( p_{ij}^{high} - p_{ij}^{low} \right) + \epsilon & \theta > 45^\circ, \end{cases}
\end{aligned}$$

where, for each parameter,  $\epsilon$  is a random variable within the range  $-\delta \leq \epsilon \leq \delta$ , with  $\delta$  being a user defined input.



**Figure 9. Probabilities of Hit/Kill Distributions**

We test the NOLH design over five different scenarios for each of the two experiments. These scenarios change the location and number of weapon types. Table 25 shows the five scenarios we test for each experiment. In the first scenario, each weapon overlaps the protected asset, located at  $(0,0,0)$ , enough that any incoming target will be within its effective range at some point. Scenario two is the same as scenario one except that there are only two available weapon types. In scenario three, the effective ranges of weapon types 1 and 2 do not overlap, thus not all targets will be within range of all weapon types. Scenario four is similar to scenario three less the availability of weapon type 3. Lastly, in scenario five all three weapon types are co-located at the protected asset.

We conduct all tests on a computer having an Intel Xeon E5-2650 v2 processor with 128 GB RAM, and we seed each technique with the same random number stream which determines the outcomes of each engagement and the success or failure of each leaker. We iterate each design point 30 times and report our results as the mean value of these replications.

**Table 25. Experimental Scenarios**

Scenario	Location			Effective Range		
	$w_1$	$w_2$	$w_3$	$w_1$	$w_2$	$w_3$
1	(10,10,0)	(-15,-15,0)	(0,0,0)	30	35	40
2	(10,10,0)	(-15,-15,0)	-	30	35	-
3	(30,0,0)	(-35,0,0)	(0,0,0)	30	35	40
4	(30,0,0)	(-35,0,0)	-	30	35	-
5	(0,0,0)	(0,0,0)	(0,0,0)	30	35	40

**One Stage Problem.**

We present the results of the performance of the single stage CRAM and the Baseline policy in Table 26. We see the average number of targets destroyed and number of protected asset hits for each algorithm for each scenario as well as the average number of each weapon type used in each scenario. As we restrict the number of weapons of types 1 and 2 to the same design points in each of the scenarios, it follows that the performance of each algorithm in scenarios 2 and 4, where only these two weapon types are available, is worse than in the other three scenarios, wherein all three weapon types are available.

We assess the performance of the CRAM by two metrics: reduction in the number of hits to the protected asset and the number of weapons used. For the first metric, we see that, in each scenario, the CRAM outperforms the Baseline policy in the number of hits allowed to the protected asset. The CRAM has an average of 0.87 hits as compared to the 2.51 hits by the Baseline over the 5 scenarios. Additionally, the number of targets destroyed is greater in the CRAM than the Baseline, with an average of 13.04 targets destroyed by the CRAM and 8.06 destroyed by the Baseline policy. Using the Kruskal-Wallis nonparametric test, we can state with 99% confidence that the CRAM destroys  $155\% \pm 4.03\%$  the number of targets that the baseline policy destroys and receives  $40.0\% \pm 5.40\%$  the hits that the baseline policy receives.

**Table 26. Single Stage Results**

Scenario	Algorithm	Targets	Protected	Weapons Used		
		Destroyed	Asset Hits	$w_1$	$w_2$	$w_3$
1	CRAM	14.01	0.61	7.65	7.20	7.73
	Baseline	8.63	2.34	9.77	9.87	10.02
2	CRAM	11.84	1.25	9.57	9.43	-
	Baseline	7.03	2.88	10.03	10.03	-
3	CRAM	13.98	0.63	7.84	7.15	8.32
	Baseline	8.45	2.43	9.82	9.88	10.01
4	CRAM	11.40	1.35	9.72	9.51	-
	Baseline	6.82	2.91	10.03	10.03	-
5	CRAM	14.20	0.54	6.35	7.62	8.37
	Baseline	9.38	2.14	9.52	10.03	10.03

The second metric marks the efficiency of the policy. The CRAM uses fewer weapons in each scenario to destroy more targets and allow fewer hits to the protected asset. On average, the CRAM uses 7.05 of each weapon type whereas the Baseline uses 8.60 of each weapon type. This indicates two improvements of the CRAM. First, it indicates that the CRAM is not conducting overkill, firing too many weapons at any one target. Second, this indicates that the CRAM assigns weapons to targets far more efficiently than the Baseline policy, saving an engagement for a weapon which will have a greater probability of kill rather than using more weapons to destroy the target. We note that an additional reason that the CRAM uses fewer weapons than the Baseline is the lack of sufficient time to allocate more weapons. Though not reported here, a reduction in the guide time parameter  $g$  increases the number of weapons used in each algorithm, yet the CRAM consistently uses fewer than the Baseline for the reasons described.

While reassigning the weapons to targets and redefining the firing order is computationally expensive, the efficiency of the heuristics defined in §3 allow for these processes to refine and improve the engagement process without sacrificing much time. As shown in Table 26, the performance of the CRAM in the single stage model



improves upon the Baseline policy in each of the metrics considered.

### Two Stage Problem.

As previously mentioned, we extend the experimentation of the one stage problem by including a second stage with two additional parameters,  $t_2^{min}$  and  $t_2^r$ , where the number of targets in the second stage,  $t_2$  is  $t_2^{min} \leq t_2 \leq t_2^{min} + t_2^r$ . As the number of targets increases for this problem, we also increase the number of weapons. We bound these parameters as follows

$$\begin{aligned} w_i & \quad 15 \leq w_i \leq 30 \\ t_2^{min} & \quad 5 \leq t_2^{min} \leq 15 \\ t_2^r & \quad 5 \leq t_2^r \leq 10. \end{aligned}$$

We present the results from the two stage experimentation in Table 27. In addition to the same measurements of effectiveness as presented in Table 26, Table 27 includes the average number of targets in the second stage,  $t_2$ . This, along with the 16 targets in the first stage, defines the average total number of targets in each scenario. However, the exact number of targets in the second stage is only known to a probability distribution and the number of weapons to save are determined in the first stage.

In observing Table 27, we can assess the performance of the CRAM as compared to the Baseline policy per the two aforementioned metrics. First, we observe that the CRAM consistently allows fewer targets to hit the protected asset than the Baseline policy. With an average of 3.72 hits, the Baseline policy allows more than the CRAM, which allows an average of 0.89 hits. Conversely, the CRAM destroys more targets than the Baseline, with averages of 23.2 and 14.2, respectively. Using the Kruskal-Wallis test, we can state with 95% confidence that the CRAM destroys  $155\% \pm 3.42\%$

**Table 27. Two Stage Results**

Scenario	Algorithm	Targets	Protected	Weapons Used			$t_2$
		Destroyed	Asset Hits	$w_1$	$w_2$	$w_3$	
1	CRAM	24.18	0.68	13.61	12.10	13.95	10.38
	Baseline	15.08	3.58	17.19	18.72	21.16	
2	CRAM	22.67	1.13	19.39	17.83	-	10.37
	Baseline	13.66	4.08	21.56	21.50	-	
3	CRAM	24.07	0.74	13.48	11.49	15.11	10.30
	Baseline	14.74	3.68	17.55	18.56	20.77	
4	CRAM	22.14	1.21	19.84	18.00	-	10.28
	Baseline	13.25	4.18	21.46	21.31	-	
5	CRAM	24.27	0.66	9.94	12.80	15.78	10.37
	Baseline	16.16	3.23	13.68	20.34	22.49	

the number of targets that the baseline policy destroys and allows  $29.8\% \pm 5.52\%$  the number of hits that the baseline policy allows.

Similar to its performance in the one stage problem, the efficiency of the CRAM in the two stage problem exceeds that of the Baseline policy. In addition to the effective assignments and prevention of overkill of any target, the CRAM utilizes the CAVE algorithm to inform the decision of how many weapons to save for the second stage. This reduces the chance of a surplus or deficit number of weapons in either stage, which reduces the total number of hits to the protected asset.

The additional computational requirement of the CAVE algorithm can be justified in the CRAM by observing its performance as compared to the Baseline policy. Though time consuming, execution of this algorithm enables an informed solution to the first problem encountered in the two stage problem: that of how many weapons to save for the second stage. With this solution, the efficient subroutines of the CRAM are able to reallocate weapons to active targets without exhausting too much time each time it observes the outcome of an engagement.

## 7.5 Conclusion

With a relatively small portion of all research into the Weapon Target Assignment problem focusing on the dynamic variant (Kline et al., 2018), there are not many models which consider a shoot-look-shoot problem within a multi stage problem. Hosein & Athans (1989) provided an early model which considers subsequent stages and targets that survived an engagement. Khosla (2001) developed a shoot-look-shoot model which considers the scheduling of engagements and can be expanded to consider multiple stages. Xin et al. (2011) developed a shoot-look-shoot model which incorporates multiple stages and proposed a solution technique that efficiently assigns available weapons to active targets in each stage. By contrast, Leboucher et al. (2013) considers only one stage but examines the continuous time position of each target and determines a firing order to engage targets before they reach the protected asset.

In this paper, we present a model which extends the continuous time model developed by Leboucher et al. (2013) into multiple stages, solving each stage in a manner similar to Xin et al. (2011). We consider differing weapon types, thus making a heterogeneous problem whose solution, in each stage, is an NP-Hard problem (Lloyd & Witsenhausen, 1986). We test the Continuous Reallocation Method (CRAM), over one and two stages by simulating scenarios in which targets follow Bézier curves to a protected asset, requiring our solution and engagements occur prior to their impact. While we evaluate the efficacy of the CRAM on two metrics (i.e., reduction in the number of hits to the protected asset and the number of weapons used), we note that there are additional metrics which may be used in future research. One such metric is the number of weapons needed so that the number of hits to the protected asset is less than some threshold. We compare the experimental results of the CRAM to those using a Baseline policy, which is a standard shoot-shoot-look policy engaging targets as they enter the effective range of a weapon.

We find that the CRAM is capable of solving this complex problem in real time and, given the parameter values defined in §4, can provide assignment and firing order policies which result in less than 1 target hitting the protected asset of the 16 incoming targets in the single stage problem and 26-31 incoming targets in the two stage problem. Through the method of simulation defined heretofore, we demonstrate its efficacy given two or three weapon types, overlapping or disjoint weapon effective ranges, and one or two stages of engagements. Using the Kruskal-Wallis nonparametric test with 95% confidence, we can state that the CRAM statistically outperforms the baseline policy in terms of the number of targets destroyed and the number of hits to the protected asset.

## VIII. Conclusion

This dissertation develops and tests real-time heuristic algorithms for the Static Weapon Target Assignment (SWTA) problem and Dynamic Weapon Target Assignment (DWTa) problem and compares results to known benchmarks or realistic baseline policies. First, a review of the literature is conducted and the state of the WTA is discussed, including modeling techniques, optimal algorithms, and heuristic algorithms. An Eminent Domain (ED) Metaheuristic is presented, which exploits the efficiency of a subroutine - the Quiz Problem (QP) Heuristic - to repeatedly solve the SWTA while denying a subset of assignments which may lead to superior solutions. A logarithmic transformation to the SWTA with tight constraints is used to improve upon the efficiency and quality of the solution to the SWTA when using the commercial solver, BARON. A heuristic which improves upon the QP Heuristic, called the Greedy Hungarian-like (GH) Heuristic, is then presented and finds superior solutions by selecting assignments which are among the best for each interceptor and each missile. The GH is used as a subroutine for the ED (identified as the GH-ED) and results for the QP, ED, GH, and GH-ED techniques are compared. A multidimensional Concave Adaptive Value Estimation (CAVE) Algorithm is developed for use in the heterogeneous two stage DWTa with a modified GH subroutine. Lastly, a Continuous Reallocation Method (CRAM) is developed, which uses the GH, GH-ED, and CAVE algorithms to solve a two stage Shoot-Look-Shoot problem.

Missile defense is a relevant problem for which a quality solution is of high complexity. Due to the probabilistic nature of incoming missiles, the success of an engagement subject to a probability of kill, the erratic nature of missile flight paths, and the uncertainty of subsequent missile strikes, which are necessary to consider in any engagement, the problem requires complex models that capture these aforementioned parameters yet whose solutions are computationally tractable. As such, simplified

models are used to develop, test, and assess real time algorithms, with which more realistic models are solved.

This dissertation is of a  $k$ -paper format and each chapter, less the introduction and conclusion, is an article that in some point of the publication process in peer reviewed journals. The scope and contributions of each article are reviewed herein.

In Chapter 2, a review of the literature on the Weapon Target Assignment (WTA) Problem is conducted in a contemporary survey. The different SWTA and DWTA models are examined and their differences and similarities are explored, demonstrating the evolution of the problem since its inception (Manne, 1958). Referencing these various models, a discussion of some of the optimal algorithms and heuristic algorithms that have been used to solve the WTA problem follows. Additionally, recent developments in the literature and alternative applications of WTA research are discussed. A metric is proposed by which the research is parsed to present among the more influential work. This is the first comprehensive survey of the WTA Problem since Cheong (1985) and includes many developments that have emerged in the years since.

In Chapter 3, the ED Metaheuristic is presented as a real-time solution technique for the SWTA and an improvement over any known solution technique found within the literature. It uses the QP Heuristic as a subroutine for this metaheuristic and compares the solution quality and required computational effort of the QP Heuristic and ED Metaheuristics to an accepted benchmark in the literature: a construction heuristic developed by Ahuja et al. (2007). The ED 1 variant is able to find solutions with two times the optimality gap of the construction heuristic, but is 2400 times faster, on average. The ED 2 variant only performs 4 times faster than the construction heuristic, but finds solutions roughly with optimality gaps 45% better than those of the construction heuristic, on average.

In Chapter 4, a systemic problem with the commercial solver, BARON, is explored. With an average false optimality rate of 21%, BARON’s unreliable performance in solving the WTA is improved using a logarithmic transformation, which results in lower bounds for which no improvements have been found. As this transformation proves to require additional computational effort, a constraint is introduced to reduce the domain, and an instance-specific parameter is used to tighten this constraint as much as possible while ensuring that no optimal solution is removed from the domain. Using this reduced-domain transformation, BARON finds superior solutions to any other technique explored and does so more efficiently than alternative approaches within BARON.

Chapter 5 develops and presents the GH heuristic, an improvement to the QP heuristic in solution quality and required computational effort. This heuristic is used as a subroutine for the ED Metaheuristic and the performance of the QP, ED, GH, and GH-ED heuristics are compared by examining their required computational effort and the quality of the solutions relative to those found by the commercial solver, BARON. It is noted that the GH-ED 1 variant finds solutions with optimality gaps 15% greater than the construction heuristic, but roughly 250 times faster, on average. The GH-ED 2 variant finds solutions with optimality gaps roughly 30% better than the construction heuristic, and was 18 times faster, on average.

In Chapter 6, a multidimensional CAVE Algorithm is developed for use in the heterogeneous two stage DWTA, using a modified GH heuristic as the subroutine. No solutions to this problem have been found in the literature to date, and this is the first technique capable of solving such an expanded model. The performance of the CAVE Algorithm is tested and the results are compared to those of a baseline policy in simulations that test the points of two nearly orthogonal Latin-hypercube designed experiments. For the smaller designed experiment, optimal policies are determined

by using a backwards induction algorithm to solve a Markov Decision Process model, which enables the assessment of the performance of the CAVE Algorithm and baseline policy. The CAVE Algorithm has an average optimality gap of 7.32% in these cases, improving upon the baseline policy by 5.46%. In the larger designed experiment, the determination of optimal policies is intractable due to the magnitude of the state space and action space, thus the CAVE Algorithm is compared to the baseline policy alone. In these larger cases, the CAVE Algorithm improves upon the baseline policy by 4.32% on average.

Chapter 7 develops a complex and realistic model with continuous time parameters which requires consideration of a Shoot-Look-Shoot firing sequence and the preservation of interceptors for a second salvo. While some models in the literature take into account one or two of the features in this model, none are as collectively realistic and complex as this model. Further, its parameters are defined in such a way as to be capable of modeling known characteristics of real missile defense systems. The CRAM, which utilizes the GH, GH-ED, and CAVE Algorithm to generate real-time solutions within the problem simulation, is developed, to which a realistic Shoot-Shoot-Look policy is compared. For the single salvo testing, the CRAM allows 66% fewer hits to the protected asset while using 18% fewer interceptors than the baseline policy. For the two salvo testing, the CRAM allows 76% fewer hits to the protected asset and uses 28% fewer interceptors than the baseline policy.

Throughout this dissertation, assumptions are made which reduced the complexity of the models under experimentation. First, in Chapters 3, 4, and 5, it is assumed that all probabilities of kill and missile destructive values are known *a-priori*. This assumption is made only in the first stage of the DWTA in Chapter 6, although it is known that the incoming missiles in the second stage are a subset of those in the first stage. In Chapter 7, these parameters are not known *a-priori*, but rather are



determined at the beginning of the simulation by analyzing the geometry of the flight path of each incoming missile. Thus, in Chapter 7, full knowledge of the flight paths of the incoming missiles in the first stage is known *a-priori*.

In Chapters 3, 4, 5, and 6, it is assumed that each interceptor has the capacity to engage each incoming missile. This assumption is relaxed in Chapter 7, wherein several scenarios in which interceptors can only engage a subset of incoming missiles are considered. Further, several scenarios are considered in Chapter 7 in which the interceptors have different earliest engagement times for the different missiles.

In Chapter 7, it is assumed that each missile has the same speed, which is constant throughout its flight path. Further, it is assumed that the time required for each engagement is constant amongst interceptor types.

Each of the solution techniques has similar problem size limitations which must be considered prior to their implementation. The ED Metaheuristic is capable of finding near optimal solutions when the ED 2 variant is used, with an average optimality gap of 1.69%. However, this variant ceases to be real-time, with average computational speed exceeding one second, for problems with more than 20 interceptors and missiles. The ED 1 variant is able to find solutions that are, on average, 3.01% above optimal, which is inferior the ED 2 in terms of solution quality, but it superior to the ED 2 in terms of required computational effort. The GH heuristic and the GH ED Metaheuristic have similar limitations. As the subroutines of the CAVE Algorithm and the CRAM are the GH heuristic and GH ED Metaheuristic, the required computational effort for each is also a limiting factor in the larger problem sizes considered.

This research has many extensions which may be explored in the future. First, the ED Metaheuristic can be improved by finding an improved method of defining the denial set. With a smaller denial set, the required computational effort of the metaheuristic improves, and if the truncation of these sets can be made in such a way

that reduces the degradation of solution quality, its application could expand.

Next, the GH Heuristic, ED Metaheuristic, and CAVE Algorithm can be used in problems unrelated to missile defense. For example, product acquisitions which have inherent probabilities of successfully achieving some goal can be modeled and solved using these techniques. If future available capital is known only to a probability distribution, the resulting problem can be modeled as the two stage DWTA, and the CAVE Algorithm can be used to efficiently obtain near-optimal solutions.

The model proposed in Chapter 7 and the corresponding solution technique, the CRAM, present a new type of model which limits the assumptions inherent to other models and captures much of the reality of a missile defense problem. As is the case with novel models, there is room for improvement in the model and its solutions that are presented in this dissertation. One extension is to disregard the knowledge of the flight paths of each missile while monitoring and predicting the flight paths based upon continuously collected information. A second extension is the consideration of sensors and their reliability on the identification of missile recognition and location.

## Bibliography

- Ahner, D. K. (2005). Planning and control of unmanned aerial vehicles in a stochastic environment. *Doctor of Philosophy dissertation, Boston University*.
- Ahner, D. K. & Parson, C. R. (2015). Optimal multi-stage allocation of weapons to targets using adaptive dynamic programming. *Optimization Letters*, 9(8), 1689–1701.
- Ahuja, R. K., Kumar, A., Jha, K. C., & Orlin, J. B. (2007). Exact and heuristic algorithms for the weapon-target assignment problem. *Operations Research*, 55(6), 1136–1146.
- Alighanbari, M. (2004). Task assignment algorithms for teams of UAVs in dynamic environments. Master’s thesis, Massachusetts Institute of Technology.
- Bertsekas, D. P. & Castañon, D. A. (1999). Rollout algorithms for stochastic scheduling problems. *Journal of Heuristics*, 5(1), 89–108.
- Bertsekas, D. P., Homer, M. L., Logan, D. A., Patek, S. D., & Sandell, N. R. (2000). Missile defense and interceptor allocation by neuro-dynamic programming. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 30(1), 42–51.
- Bertuccelli, L. F. & How, J. P. (2011). Active exploration in robust unmanned vehicle task assignment. *Journal of Aerospace Computing, Information, and Communication*, 8(8), 250–268.
- Boardman, N. T., Lunday, B. J., & Robbins, M. J. (2017). Heterogeneous surface-to-air missile defense battery location: a game theoretic approach. *Journal of Heuristics*, 23(6), 417–447.
- Bogdanowicz, Z. & Coleman, N. (2007). Sensor-target and weapon-target pairings based on auction algorithm. In *Proceedings of the 11th WSEAS International Conference on Applied Mathematics*, (pp. 92–96). World Scientific and Engineering Academy and Soc.(WSEAS) Stevens Point, WI.
- Bogdanowicz, Z. R. (2012). Advanced input generating algorithm for effect-based weapon-target pairing optimization. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 42(1), 276–280.
- Bogdanowicz, Z. R., Tolano, A., Patel, K., & Coleman, N. P. (2013). Optimization of weapon-target pairings based on kill probabilities. *IEEE Transactions on Cybernetics*, 43(6), 1835–1844.
- Burr, S. A., Falk, J. E., & Karr, A. F. (1985). Integer prim-read solutions to a class of target defense problems. *Operations Research*, 33(4), 726–745.

- Çetin, E. (2007). A queuing theoretical model for anticancer tool selection. In *International Mathematical Forum*, volume 2, (pp. 2675–2685).
- Çetin, E. & Esen, S. T. (2006). A weapon–target assignment approach to media allocation. *Applied Mathematics and Computation*, 175(2), 1266–1275.
- Chang, S.-c., James, R. M., & Shaw, J. J. (1987). Assignment algorithm for kinetic energy weapons in boost phase defence. In *26th IEEE Conference on Decision and Control, 1987*, volume 26, (pp. 1678–1683). IEEE.
- Chen, J., Xin, B., Peng, Z., Dou, L., & Zhang, J. (2009). Evolutionary decision-makings for the dynamic weapon-target assignment problem. *Science in China Series F: Information Sciences*, 52(11), 2006–2018.
- Cheong, C. K. (1985). Survey of investigations into the missile allocation problem. Master’s thesis, Naval Postgraduate School.
- Day, R. H. (1966). Allocating weapons to target complexes by means of nonlinear programming. *Operations Research*, 14(6), 992–1013.
- denBroeder, G., Ellison, R., & Emerling, L. (1959). On optimum target assignments. *Operations Research*, 7(3), 322–326.
- Eckler, A. R. & Burr, S. A. (1972). Mathematical models of target coverage and missile allocation. Technical report, Military Operations Research Society Alexandria VA.
- Esen, Ö., Çetin, E., & Esen, S. T. (2008). A mathematical immunochemoradiotherapy model: A multiobjective approach. *Nonlinear Analysis: Real World Applications*, 9(2), 511–517.
- Fu, T.-p., Liu, Y.-s., & Chen, J.-h. (2006). Improved genetic and ant colony optimization algorithm for regional air defense wta problem. In *First International Conference on Innovative Computing, Information and Control, 2006*, volume 1, (pp. 226–229). IEEE.
- Gelenbe, E., Timotheou, S., & Nicholson, D. (2010). Fast distributed near-optimum assignment of assets to tasks. *The Computer Journal*, 53(9), 1360–1369.
- Godfrey, G. A. & Powell, W. B. (2001). An adaptive, distribution-free algorithm for the newsvendor problem with censored demands, with applications to inventory and distribution. *Management Science*, 47(8), 1101–1112.
- Golany, B., Goldberg, N., & Rothblum, U. G. (2015). Allocating multiple defensive resources in a zero-sum game setting. *Annals of Operations Research*, 225(1), 91–109.

- Golany, B., Goldberg, N., & Rothblum, U. G. (2017). A two-resource allocation algorithm with an application to large-scale zero-sum defensive games. *Computers & Operations Research*, 78, 218–229.
- Gülpınar, N., Çanakoğlu, E., & Branke, J. (2018). Heuristics for the stochastic dynamic task-resource allocation problem with retry opportunities. *European Journal of Operational Research*, 266(1), 291–303.
- Hill, R. R. & Pohl, E. A. (2010). Heuristics and their use in military modeling. *Wiley Encyclopedia of Operations Research and Management Science*.
- Hosein, P. A. (1989). A class of dynamic nonlinear resource allocation problems. Technical report, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology.
- Hosein, P. A. & Athans, M. (1989). The dynamic weapon-target assignment problem. Technical report, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology.
- Hosein, P. A., Walton, J. T., & Athans, M. (1988). Dynamic weapon-target assignment problems with vulnerable  $c^2$  nodes. Technical report, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology.
- Huang, S.-c. & LI, W.-m. (2005). Research of ant colony algorithm for solving target assignment problem [j]. *Systems Engineering and Electronics*, 1, 023.
- Jian, W. & Chen, C. (2015). Sensor-weapon joint management based on improved genetic algorithm. In *34th Chinese Control Conference (CCC), 2015*, (pp. 2738–2742). IEEE.
- Johansson, F. & Falkman, G. (2009). An empirical investigation of the static weapon-target allocation problem. In *Proceedings of the 3rd Skövde Workshop on Information Fusion Topics*, (pp. 63–67).
- Johansson, F. & Falkman, G. (2010). Real-time allocation of defensive resources to rockets, artillery, and mortars. In *13th Conference on Information Fusion (FUSION)*, (pp. 1–8). IEEE.
- Julstrom, B. A. (2009). String-and permutation-coded genetic algorithms for the static weapon-target assignment problem. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, (pp. 2553–2558). ACM.
- Karasakal, O. (2008). Air defense missile-target allocation models for a naval task group. *Computers & Operations Research*, 35(6), 1759–1770.
- Khosla, D. (2001). Hybrid genetic approach for the dynamic weapon-target allocation problem. In *Proceedings of SPIE*, volume 4396, (pp. 248–263). SHE.

- Kline, A. (2017). Real-time heuristic algorithms for the static weapon-target assignment problem. Master's thesis, Air Force Institute of Technology.
- Kline, A., Ahner, D., & Hill, R. (2018). The weapon-target assignment problem. Technical report, Air Force Institute of Technology.
- Kline, A., Ahner, D., & Lunday, B. (2017). A heuristic and metaheuristic approach to the static weapon target assignment problem. Technical report, Air Force Institute of Technology.
- Kline, A., Ahner, D., & Pachter, M. (2017). A greedy hungarian algorithm for the weapon-target assignment problem. Technical report, Air Force Institute of Technology.
- Kline, A., Ahner, D., & Parson, C. (2018). Solving the heterogeneous multi-stage weapon target assignment problem with adaptive dynamic programming. Technical report, Air Force Institute of Technology.
- Kolitz, S. E. (1988). Analysis of a maximum marginal return assignment algorithm. In *Proceedings of the 27th IEEE Conference on Decision and Control*, (pp. 2431–2436). IEEE.
- Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics*, 2(1-2), 83–97.
- Kwon, O., Kang, D., Lee, K., & Park, S. (1999). Lagrangian relaxation approach to the targeting problem. *Naval Research Logistics*, 46(6), 640–653.
- Leboucher, C., Shin, H.-S., Siarry, P., Chelouah, R., Le Méné, S., & Tsourdos, A. (2013). A two-step optimisation method for dynamic weapon target assignment problem. In *Recent Advances on Meta-Heuristics and Their Application to Real Scenarios*. InTech.
- Lee, M.-Z. (2010). Constrained weapon–target assignment: Enhanced very large scale neighborhood search algorithm. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(1), 198–204.
- Lee, Z.-J. & Lee, C.-Y. (2005). A hybrid search algorithm with heuristics for resource allocation problem. *Information Sciences*, 173(1), 155–167.
- Lee, Z.-J., Lee, C.-Y., & Su, S.-F. (2002). An immunity-based ant colony optimization algorithm for solving weapon–target assignment problem. *Applied Soft Computing*, 2(1), 39–47.
- Lee, Z.-J. & Lee, W.-L. (2003). A hybrid search algorithm of ant colony optimization and genetic algorithm applied to weapon-target assignment problems. In *IDEAL*, volume 2003, (pp. 278–285). Springer.

- Lee, Z.-J., Su, S.-F., & Lee, C.-Y. (2002). A genetic algorithm with domain knowledge for weapon-target assignment problems. *Journal of the Chinese Institute of Engineers*, 25(3), 287–295.
- Lee, Z.-J., Su, S.-F., & Lee, C.-Y. (2003). Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 33(1), 113–121.
- Lemus, F. & David, K. (1963). An optimum allocation of different weapons to a target complex. *Operations Research*, 11(5), 787–794.
- Li, J., Chen, J., Xin, B., & Chen, L. (2017). Efficient multi-objective evolutionary algorithms for solving the multi-stage weapon target assignment problem: A comparison study. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 435–442). IEEE.
- Li, J., Chen, J., Xin, B., & Dou, L. (2015). Solving multi-objective multi-stage weapon target assignment problem via adaptive NSGA-II and adaptive MOEA/D: A comparison study. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 3132–3139). IEEE.
- Li, P., Wu, L., & Lu, F. (2009). A mutation-based ga for weapon-target allocation problem subject to spatial constraints. In *International Workshop on Intelligent Systems and Applications*, (pp. 1–4). IEEE.
- Li, Y., Kou, Y., & Li, Z. (2018). An improved nondominated sorting Genetic Algorithm III Method for solving multiobjective weapon-target assignment Part I: The value of fighter combat. *International Journal of Aerospace Engineering*, 2018.
- Li, Y., Kou, Y., Li, Z., Xu, A., & Chang, Y. (2017). A modified pareto ant colony optimization approach to solve biobjective weapon-target assignment problem. *International Journal of Aerospace Engineering*, 2017.
- Lloyd, S. P. & Witsenhausen, H. S. (1986). Weapons allocation is np-complete. In *1986 Summer Computer Simulation Conference*, (pp. 1054–1058).
- Lu, H., Zhang, H., Zhang, X., & Han, R. (2006). An improved genetic algorithm for target assignment, optimization of naval fleet air defense. In *The Sixth World Congress on Intelligent Control and Automation*, volume 1, (pp. 3401–3405). IEEE.
- Madni, A. M. & Andrecut, M. (2009). Efficient heuristic approaches to the weapon-target assignment problem. *AIAA Journal of Aerospace Computing, Information, and Communication*, 6, 405–414.
- Malcolm, W. P. (2004). On the character and complexity of certain defensive resource allocation problems. Technical report.

- Manne, A. S. (1958). A target-assignment problem. *Operations Research*, 6(3), 346–351.
- Matlin, S. (1970). A review of the literature on the missile-allocation problem. *Operations Research*, 18(2), 334–373.
- Metler, W. A., Preston, F. L., & Hofmann, J. (1990). A suite of weapon assignment algorithms for a sdi mid-course battle manager. Technical report, NAVAL RESEARCH LAB WASHINGTON DC.
- Murphey, R. A. (2000). An approximate algorithm for a weapon target assignment stochastic program. In *Approximation and Complexity in Numerical Optimization* (pp. 406–421). Springer.
- Neumaier, A., Shcherbina, O., Huyer, W., & Vinkó, T. (2005). A comparison of complete global optimization solvers. *Mathematical Programming*, 103(2), 335–356.
- Onay, O. (2016). A mathematical approach to neuromarketing: A weapon target assignment model. *International Journal of Academic Research in Business and Social Sciences*, 6(1), 164–173.
- Rosenberger, J. M., Hwang, H. S., Pallerla, R. P., Yucel, A., Wilson, R. L., & Brungardt, E. G. (2005). The generalized weapon target assignment problem. Technical report, Texas Univ At Arlington.
- Shalumov, V. & Shima, T. (2017). Weapon–target-allocation strategies in multiagent target–missile–defender engagement. *Journal of Guidance, Control, and Dynamics*, 40(10), 2452–2464.
- Shan, X. & Zhuang, J. (2013). Hybrid defensive resource allocations in the face of partially strategic attackers in a sequential defender–attacker game. *European Journal of Operational Research*, 228(1), 262–272.
- Shang, G. (2003). Ant colony algorithm for weapon-target assignment problem [j]. *Computer Engineering and Applications*, 3, 025.
- Shang, G. (2008). Solving weapon-target assignment problems by a new ant colony algorithm. In *International Symposium on Computational Intelligence and Design*, volume 1, (pp. 221–224). IEEE.
- Shang, G., Zaiyue, Z., Xiaoru, Z., & Cungen, C. (2007). Immune genetic algorithm for weapon-target assignment problem. In *Workshop on Intelligent Information Technology Application*, (pp. 145–148). IEEE.
- Sikanen, T. (2008). Solving weapon target assignment problem with dynamic programming. Technical Report 55670.



- Soland, R. M. (1987). Optimal terminal defense tactics when several sequential engagements are possible. *Operations Research*, 35(4), 537–542.
- Su, M., QIAN, H., & WANG, X.-f. (2008). Immune memory-based ant colony algorithm for weapon-target assignment solution. *Computer Engineering*, 4, 078.
- Tawarmalani, M. & Sahinidis, N. V. (2004). Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99(3), 563–591.
- Tokgöz, A. & Bulkan, S. (2013). Weapon target assignment with combinatorial optimization techniques. *IJARAI International Journal of Advanced Research in Artificial Intelligence*, 2(7), 39–50.
- Wacholder, E. (1989). A neural network-based optimization algorithm for the static weapon-target assignment problem. *ORSA Journal on Computing*, 1(4), 232–246.
- Wu, L., Wang, H.-y., Lu, F.-x., & Jia, P. (2008). An anytime algorithm based on modified ga for dynamic weapon-target allocation problem. In *IEEE Congress on Evolutionary Computation*, (pp. 2020–2025). IEEE.
- Xin, B., Chen, J., Peng, Z., Dou, L., & Zhang, J. (2011). An efficient rule-based constructive heuristic to solve dynamic weapon-target assignment problem. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41(3), 598–606.
- Xin, B., Chen, J., Zhang, J., Dou, L., & Peng, Z. (2010). Efficient decision makings for dynamic weapon-target assignment by virtual permutation and tabu search heuristics. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(6), 649–662.
- Xin, B., Wang, Y., & Chen, J. (2018). An efficient marginal-return-based constructive heuristic to solve the sensor-weapon-target assignment problem. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, (99), 1–12.
- Yanxia, W., Longjun, Q., Zhi, G., & Lifeng, M. (2008). Weapon target assignment problem satisfying expected damage probabilities based on ant colony algorithm. *Journal of Systems Engineering and Electronics*, 19(5), 939–944.
- Zeng, X., Zhu, Y., Nan, L., Hu, K., Niu, B., & He, X. (2006). Solving weapon-target assignment problem using discrete particle swarm optimization. In *The Sixth World Congress on Intelligent Control and Automation*, volume 1, (pp. 3562–3565). IEEE.
- Zhihua, S., Zhu, F., & Duolin, Z. (2009). A heuristic genetic algorithm for solving constrained weapon-target assignment problem. In *IEEE International Conference on Intelligent Computing and Intelligent Systems*, volume 1, (pp. 336–341). IEEE.

Zi-fen, L., Xiang-min, L., Jin-jin, D., Jin-zhu, C., & Feng-xia, Z. (2011). Sensor-weapon-target assignment based on improved swt-opt algorithm. In *2011 IEEE 2nd International Conference on Computing, Control and Industrial Engineering (CCIE)*, volume 2, (pp. 25–28). IEEE.

<b>REPORT DOCUMENTATION PAGE</b>					<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. <b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b>						
<b>1. REPORT DATE</b> (DD-MM-YYYY) 20-12-2018		<b>2. REPORT TYPE</b> Doctoral Dissertation		<b>3. DATES COVERED</b> (From — To) March 2016 — October 2018		
<b>4. TITLE AND SUBTITLE</b>  REAL-TIME HEURISTICS AND METAHEURISTICS FOR STATIC AND DYNAMIC WEAPON TARGET ASSIGNMENTS				<b>5a. CONTRACT NUMBER</b>  <b>5b. GRANT NUMBER</b>  <b>5c. PROGRAM ELEMENT NUMBER</b>		
<b>6. AUTHOR(S)</b>  Alexander G. Kline, CPT, USA				<b>5d. PROJECT NUMBER</b>  <b>5e. TASK NUMBER</b>  <b>5f. WORK UNIT NUMBER</b>		
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>  AFIT-ENS-DS-18-D-016		
<b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> STRATCOM/JWAC Attn: Jill Morrissett, J902 4048 Higley Rd Dahlgren, VA 22448 jill.s.morrissett.civ@mail.mil				<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b>  JWAC  <b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b>		
<b>12. DISTRIBUTION / AVAILABILITY STATEMENT</b> DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.						
<b>13. SUPPLEMENTARY NOTES</b>  This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.						
<b>14. ABSTRACT</b> The problem of targeting and engaging individual missiles (targets) with an arsenal of interceptors (weapons) is known as the weapon target assignment (WTA) problem. This problem has been well-researched since the seminal work in 1958. There are two distinct categories of the WTA: static (SWTA) and dynamic (DWTa). The SWTA considers a single instance in which a known number of incoming missiles is to be engaged with a finite number of interceptors. By contrast, the DWTa considers either follow on engagement(s) should the first engagement(s) fail, a subsequent salvo of incoming missiles, or both. This research seeks to define and solve a realistic dynamic model. First, we develop assignment heuristics and metaheuristics which provide rapid near-optimal solutions to the SWTA. Next, we develop a technique capable of solving the heterogeneous DWTa. Lastly, we develop a model which realistically considers erratic flight paths of incoming missiles and solves the model within a simulation. Collectively, this work extends the research of missile defense into practical application more so than currently can be found within the literature.						
<b>15. SUBJECT TERMS</b>  Weapon Target Assignment Problem, Static, Dynamic, Heuristic, Metaheuristic, Dynamic Programming, Survey						
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b>		<b>18. NUMBER OF PAGES</b>	
<b>a. REPORT</b>  U	<b>b. ABSTRACT</b>  U	<b>c. THIS PAGE</b>  U	U		162	
			<b>19a. NAME OF RESPONSIBLE PERSON</b> Dr. Darryl K. Ahner, AFIT/ENS			
			<b>19b. TELEPHONE NUMBER</b> (include area code) 937-255-6565 x4708; darryl.ahner@afit.edu			